

WRDC-TR-90-8007
Volume VIII
Part 8

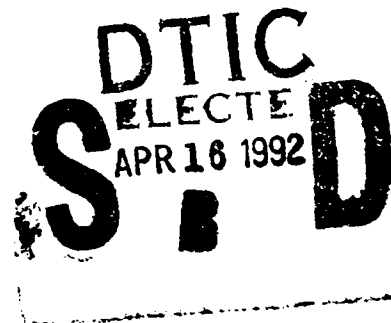
AD-A248 918



INTEGRATED INFORMATION SUPPORT SYSTEM (IISS)
Volume VIII - User Interface Subsystem
Part 8 - Forms Processor Product Specification

S. Barker

Control Data Corporation
Integration Technology Services
2970 Presidential Drive
Fairborn, OH 45324-6209



September 1990

Final Report for Period 1 April 1987 - 31 December 1990

Approved for Public Release; Distribution is Unlimited

92 4 14 046

MANUFACTURING TECHNOLOGY DIRECTORATE
WRIGHT RESEARCH AND DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6533

92-09613



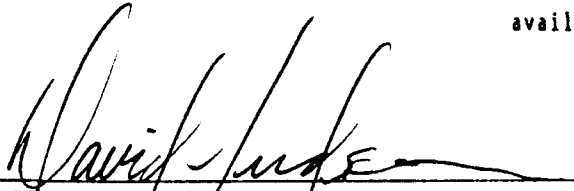
~~92 4 14 046~~

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, regardless whether or not the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data. It should not, therefore, be construed or implied by any person, persons, or organization that the Government is licensing or conveying any rights or permission to manufacture, use, or market any patented invention that may in any way be related thereto.

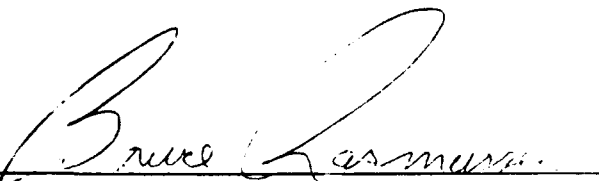
This technical report has been reviewed and is approved for publication.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations


DAVID L. JUDSON, Project Manager
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

25 July 91
DATE

FOR THE COMMANDER:


BRUCE A. RASMUSSEN, Chief
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

25 July 91
DATE

If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify WRDC/MTI, Wright-Patterson Air Force Base, OH 45433-6533 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified		1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release; Distribution is Unlimited.	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE			
4. PERFORMING ORGANIZATION REPORT NUMBER(S) PS 620344200		5. MONITORING ORGANIZATION REPORT NUMBER(S) WRDC-TR- 90-8007 Vol. VIII, Part 8	
6a. NAME OF PERFORMING ORGANIZATION Control Data Corporation; Integration Technology Services	6b. OFFICE SYMBOL (if applicable) WRDC/MTI	7a. NAME OF MONITORING ORGANIZATION WRDC/MTI	
6c. ADDRESS (City, State, and ZIP Code) 2970 Presidential Drive Fairborn, OH 45324-6209		7b. ADDRESS (City, State, and ZIP Code) WPAFB, OH 45433-6533	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Wright Research and Development Center, Air Force Systems Command, USAF	8b. OFFICE SYMBOL (if applicable) WRDC/MTI	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUM. F33600-87-C-0464	
8c. ADDRESS (City, State, and ZIP Code) Wright-Patterson AFB, Ohio 45433-6533		10. SOURCE OF FUNDING NOS.	
11. TITLE (Include Security Classification) See block 19		PROGRAM ELEMENT NO. 78011F	PROJECT NO. 595600
		TASK NO. F95600	WORK UNIT NO. 20950607
12. PERSONAL AUTHOR(S) Structural Dynamics Research Corporation: Barker, S., et al.			
13a. TYPE OF REPORT Final Report	13b. TIME COVERED 4 / 1 / 87 - 12 / 31 / 90	14. DATE OF REPORT (Yr., Mo., Day) 1990 September 30	15. PAGE COUNT 387
16. SUPPLEMENTARY NOTES WRDC/MTI Project Priority 6203			
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify block no.)	
FIELD	GROUP	SUB GR.	
1308	0905		
19. ABSTRACT (Continue on reverse if necessary and identify block number) This specification establishes the detailed design of the Form Processor (FP) computer program. BLOCK 11: INTEGRATED INFORMATION SUPPORT SYSTEM Vol VIII -User Interface Subsystem Part 8 - Forms Processor Product Specification			
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED x SAME AS RPT. DTIC USERS		21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL David L. Judson	22b. TELEPHONE NO. (Include Area Code) (513) 255-7371	22c. OFFICE SYMBOL WRDC/MTI	

FOREWORD

This technical report covers work performed under Air Force Contract F33600-87-C-0464, DAPro Project. This contract is sponsored by the Manufacturing Technology Directorate, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. It was administered under the technical direction of Mr. Bruce A. Rasmussen, Branch Chief, Integration Technology Division, Manufacturing Technology Directorate, through Mr. David L. Judson, Project Manager. The Prime Contractor was Integration Technology Services, Software Programs Division, of the Control Data Corporation, Dayton, Ohio, under the direction of Mr. W. A. Osborne. The DAPro Project Manager for Control Data Corporation was Mr. Jimmy P. Maxwell.

The DAPro project was created to continue the development, test, and demonstration of the Integrated Information Support System (IISS). The IISS technology work comprises enhancements to IISS software and the establishment and operation of IISS test bed hardware and communications for developers and users.

The following list names the Control Data Corporation subcontractors and their contributing activities:

<u>SUBCONTRACTOR</u>	<u>ROLE</u>
Control Data Corporation	Responsible for the overall Common Data Model design development and implementation, IISS integration and test, and technology transfer of IISS.
D. Appleton Company	Responsible for providing software information services for the Common Data Model and IDEF1X integration methodology.
ONTEK	Responsible for defining and testing a representative integrated system base in Artificial Intelligence techniques to establish fitness for use.
Simpact Corporation	Responsible for Communication development.
Structural Dynamics Research Corporation	Responsible for User Interfaces, Virtual Terminal Interface, and Network Transaction Manager design, development, implementation, and support.
Arizona State University	Responsible for test bed operations and support.

TABLE OF CONTENTS

		<u>Page</u>
SECTION 1.0	SCOPE	1-1
1.1	Identification	1-1
1.2	Functional Summary	1-1
SECTION 2.0	DOCUMENTS	2-1
2.1	Reference Documents	2-1
2.2	Terms and Abbreviations	2-2
SECTION 3.0	REQUIREMENTS	3-1
3.1	Structural Description	3-1
3.2	Functional Flow	3-1
3.3	Interfaces	3-2
3.3.1	Application	3-4
3.3.2	Virtual Terminal	3-5
3.3.3	Forms Definition Language	3-5
3.4	Program Interrupts	3-5
3.5	Timing and Sequencing Description	3-5
3.6	Special Control Features	3-5
3.7	Storage Allocation	3-5
3.7.1	Data Base Definition	3-5
3.7.1.1	File Descriptions	3-6
3.8	Object Code Creation	3-7
3.9	Adaptation Data	3-7
3.10	Detailed Design Description	3-7
3.10.1	Main Program List	3-7
3.10.2	Module List	3-10
3.10.3	External Routines List	3-18
3.10.4	Include File List	3-21
3.10.5	Where Include File Used List	3-23
3.10.6	Where External Routine Used List	3-43
3.10.7	Main Program Parts List	3-62
3.10.8	Module Documentation	3-72
3.10.9	Include File Description	3-308
3.10.10	Hierarchy Chart	3-328
3.11	Program Listings Comments	3-375
SECTION 4.0	QUALITY ASSURANCE PROVISIONS	4-1
4.1	Introduction and Definitions	4-1
4.2	Computer Programming and Test Evaluation	4-1

List of Illustrations

<u>Figure</u>	<u>Title</u>	<u>Page</u>
3-1	Form Processor Data Flow	3-1
3-2	FP Stand Alone (non IISS environment)	3-3
3-3	FP in IISS Environment	3-4

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



SECTION 1

SCOPE

1.1 Identification

This specification establishes the detailed design of a computer program identified as the Form Processor, hereinafter referred to as the FP. The FP is one configuration item of the Integrated Information Support System (IISS) User Interface (UI).

1.2 Functional Summary

One of the objectives of the IISS testbed is to allow applications to be run from a wide variety of terminals using formatted screens for input and output of application data. Instead of the application programs having to contain terminal dependent code to send/receive formatted screens to/from various types of terminals and to perform terminal control functions, the program may use the set of callable execution time routines of the FP.

The major functions of the FP are:

1. Opening and displaying a form, a template defining fields and their attributes.
2. Placing data into a form and/or into a form message line.
3. Sending the form to the terminal.
4. Reading the data from the terminal.
5. Stacking/replacing forms currently open for the application program.
6. IISS logon processing.
7. NTM message processing.
8. Window management processing.

SECTION 2

DOCUMENTS

2.1 Reference Documents

- [1] Structural Dynamics Research Corporation, Application Interface Product Specification, PS 620144700 , 1 November 1985.
- [2] Structural Dynamics Research Corporation, Forms Driven Form Editor Product Specification, PS 620144402 , 1 November 1985.
- [3] Structural Dynamics Research Corporation, Forms Language Compiler Product Specification, PS 620144401 , 1 November 1985.
- [4] Structural Dynamics Research Corporation, Text Editor Product Specification, PS 620144600 , 1 November 1985.
- [5] Structural Dynamics Research Corporation, Rapid Application Generator Product Specification, PS 620144502 , 1 November 1985.
- [6] Structural Dynamics Research Corporation, Report Writer Product Specification, PS 620144501 , 1 November 1985.
- [7] Structural Dynamics Research Corporation, User Interface Services Product Specification, PS 620144100 , 1 November 1985.
- [8] Structural Dynamics Research Corporation, Virtual Terminal Product Specification, PS 620144300 , 1 November 1985.
- [9] Structural Dynamics Research Corporation, Form Processor Development Specification, DS 620144200B, 1 November 1985.
- [10] Structural Dynamics Research Corporation, Form Processor User Manual, UM 620144200B, 1 November 1985.
- [11] Structural Dynamics Research Corporation, Form Processor Unit Test Plan, UTP620144200 , 1 November 1985.

2.2 Terms and Abbreviations

American Standard Code for Information Interchange: (ASCII), the character set defined by ANSI X3.4 and used by most computer vendors.

Application Interface: (AI), subset of the IISS User Interface that consists of the callable routines that are linked with applications that use the Form Processor or Virtual Terminal. The AI enables applications to be hosted on computers other than the host of the User Interface.

Application Process: (AP), a cohesive unit of software that can be initiated as a unit to perform some function or functions.

Attribute: field characteristic such as blinking, highlighted, black, etc. and various other combinations. Background attributes are defined for forms or windows only. Foreground attributes are defined for items. Attributes may be permanent, i.e., they remain the same unless changed by the application program, or they may be temporary, i.e., they remain in effect until the window is redisplayed.

Common Data Model: (CDM), IISS subsystem that describes common data application process formats, form definitions, etc. of the IISS and includes conceptual schema, external schemas, internal schemas, and schema transformation operators.

Computer Program Configuration Item: (CPCI), an aggregation of computer programs or any of their discrete portions, which satisfies an end-use function.

Conceptual Schema: (CS), the standard definition used for all data in the CDM. It is based on IDEF1 information modelling.

Current Cursor Position: the position of the cursor before an edit command or function is issued in the text editor.

Cursor Position: the position of the cursor after any command is issued.

Device Drivers: (DD), software modules written to handle I/O for a specific kind of terminal. The modules map terminal specific commands and data to a neutral format. Device Drivers are part of the UI Virtual Terminal.

Display List: is similar to the open list, except that it contains only those forms that have been added to the screen and are currently displayed on the screen.

Display Size: the number of lines used in the edit area.

Extended Binary Coded Decimal Interchange Code: (EBCDIC), the character set used by a few computer vendors (notably IBM) instead of ASCII.

External Schema: (ES), an application's view of the CDM's conceptual schema.

Field: two dimensional space on a terminal screen.

Field Pointer: indicates the ITEM which contains the current cursor position.

Form: structured view which may be imposed on windows or other forms. A form is composed of fields. These fields may be defined as forms, items, and windows.

Form Definition: (FD), forms definition language after compilation. It is read at runtime by the Form Processor.

Forms Definition Language: (FDL), the language in which electronic forms are defined.

Forms Driven Form Editor: (FD FE), subset of the FE which consists of a forms driven application used to create Form Definition files interactively.

Form Editor: (FE), subset of the IISS User Interface that is used to create definitions of forms. The FE consists of the Forms Driven Form Editor and the Forms Language Compiler.

Form Hierarchy: a graphic representation of the way in which forms, items and windows are related to their parent form.

Forms Language Compiler: (FLAN), subset of the FE that consists of a batch process that accepts a series of forms definition language statements and produces form definition files as output.

Form Processor: (FP), subset of the IISS User Interface that consists of a set of callable execution time routines available to an application program for form processing.

Form Processor Text Editor: (FPTE), subset of the Form Processor that consists of software modules that provide text editing capabilities to all users of applications that use the Form Processor.

Integrated Information Support System: (IISS), a test computing environment used to investigate, demonstrate and test the concepts of information management and information integration in the context of Aerospace Manufacturing. The IISS addresses the problems of integration of data resident on heterogeneous data bases supported by heterogeneous computers interconnected via a Local Area Network.

Item: non-decomposable area of a form in which hard-coded descriptive text may be placed and the only defined areas where user data may be input/output.

Logical Device: a conceptual device which to an application

is indistinguishable from a physical device and is then mapped to part or all of a physical device.

Message: descriptive text which may be returned in the standard message line on the terminal screen. They are used to warn of errors or provide other user information.

Message Line: a line on the terminal screen that is used to display messages.

Network Transaction Manager: (NTM), IISS subsystem that performs the coordination, communication and housekeeping functions required to integrate the Application Processes and System Services resident on the various hosts into a cohesive system.

Open List: a list of all the forms that have been and are currently open for an application process.

Operating System: (OS), software supplied with a computer which allows it to supervise its own operations and manage access to hardware facilities such as memory and peripherals.

Page: instance of forms in windows that are created whenever a form is added to a window.

Paging and Scrolling: a method which allows a form to contain more data than can be displayed with provisions for viewing any portion of the data buffer.

Physical Device: a hardware terminal.

Presentation Schema: (PS), may be equivalent to a form. It is the view presented to the user of the application.

Previous Cursor Position: the position of the cursor when the previous edit command was issued.

Qualified Name: the name of a form, item or window preceded by the hierarchy path so that it is uniquely identified.

Report Definition Language: an extension of the Forms Definition Language that includes retrieval and calculation of database information and is used to define reports.

Subform: a form that is used within another form.

User Data: data which is either input by the user or output by the application programs to items.

User Interface: (UI), IISS subsystem that controls the user's terminal and interfaces with the rest of the system. The UI consists of two major subsystems: the User Interface Development System (UIDS) and the User Interface Management System (UIMS).

User Interface Development System: (UIDS), collection of

IISS User Interface subsystems that are used by applications programmers as they develop IISS applications. The UIDS includes the Form Editor and the Application Generator.

User Interface Management System: (UIMS), the runtime UI. It consists of the Form Processor, Virtual Terminal, Application Interface, the User Interface Services and the Text Editor.

User Interface Monitor: (UIM), part of the Form Processor that handles messaging between the NTM and the UI. It also provides authorization checks and initiates applications.

User Interface/Virtual Terminal Interface: (UI/VTI), another name for the User Interface.

Virtual Terminal: (VT), subset of the IISS User Interface that performs the interfacing between different terminals and the UI. This is done by defining a specific set of terminal features and protocols which must be supported by the UI software which constitutes the virtual terminal definition. Specific terminals are then mapped against the virtual terminal software by specific software modules written for each type of real terminal supported.

Virtual Terminal Interface: (VTI), the callable interface to the VT.

Window: dynamic area of a terminal screen on which predefined forms may be placed at run time.

Window Manager: a facility which allows the following to be manipulated: size and location of windows, the device on which an application is running, the position of a form within a window. It is part of the Form Processor.

SECTION 3

REQUIREMENTS

3.1 Structural Description

The overall structure of the Form Processor is based on a User Interface Monitor which interprets the Application Interface messages in order to call the appropriate Form Processor routine. All the FP callable routines are at the same level in the hierarchical structure of the FP CPCI.

3.2 Functional Flow

Figure 3-1 is a data flow diagram of the Form Processor.

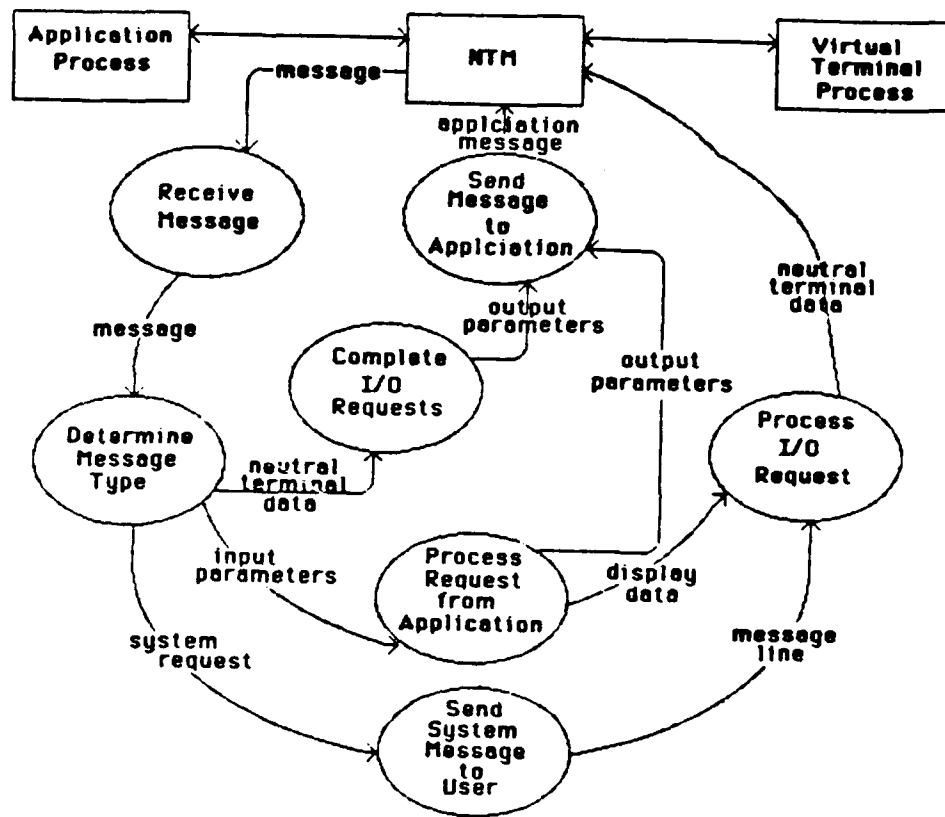


Figure 3-1 Form Processor Data Flow

3.3 Interfaces

The FP interfaces with the NTM through its UIM, with the AI through its UIM and callable routines, and with the VT through the VT Form Processor callable routines. In addition, two different methods of interfacing to an AP are supported: stand alone, when the AP may link directly to the FP and no NTM is being used, or the IISS environment, when the AP may link to the Application interface (AI) routines. In either environment, the application programs use the exact same interface to the FP. Linking to the FO directly is simpler and more efficient than using the AI but does not support the use of an NTM; direct linking is most useful for testing purposes.

The UIM part of the FP only exists when the NTM is being used. The UIM receives the message formatted by the AI and translates it into the appropriate FP callable routine to permit the sending or receiving of the forms. The FP routines then interface with the VT by translating an application request into the appropriate VT command when input/output is necessary.

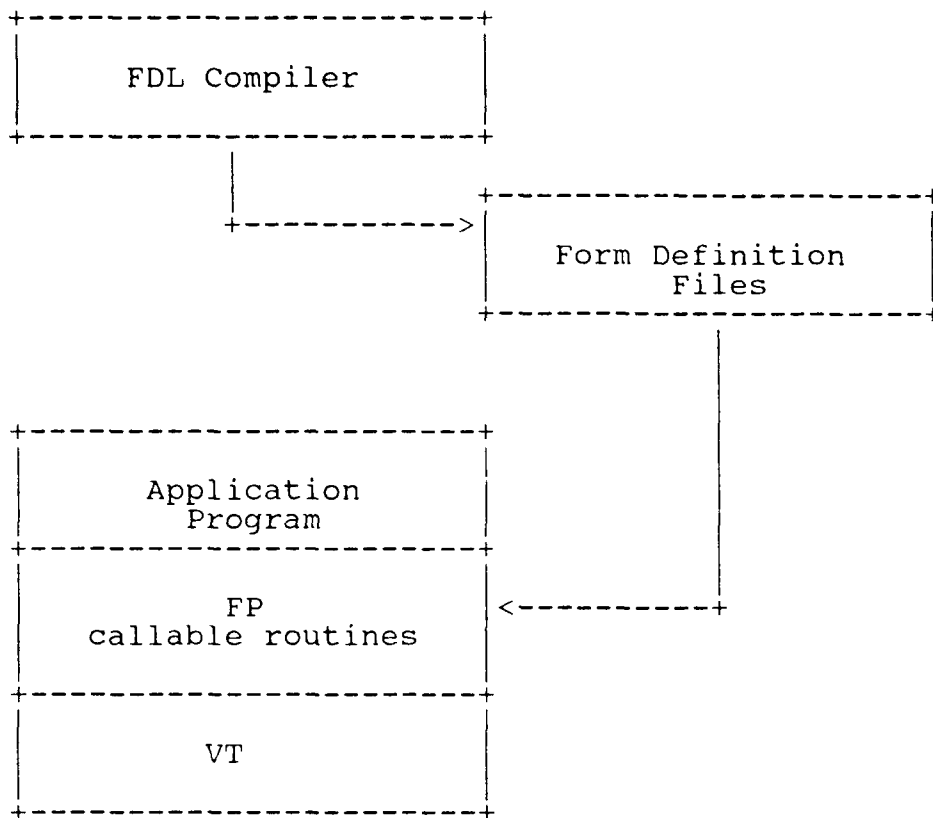


Figure 3-2 FP Stand Alone (non IISS environment)

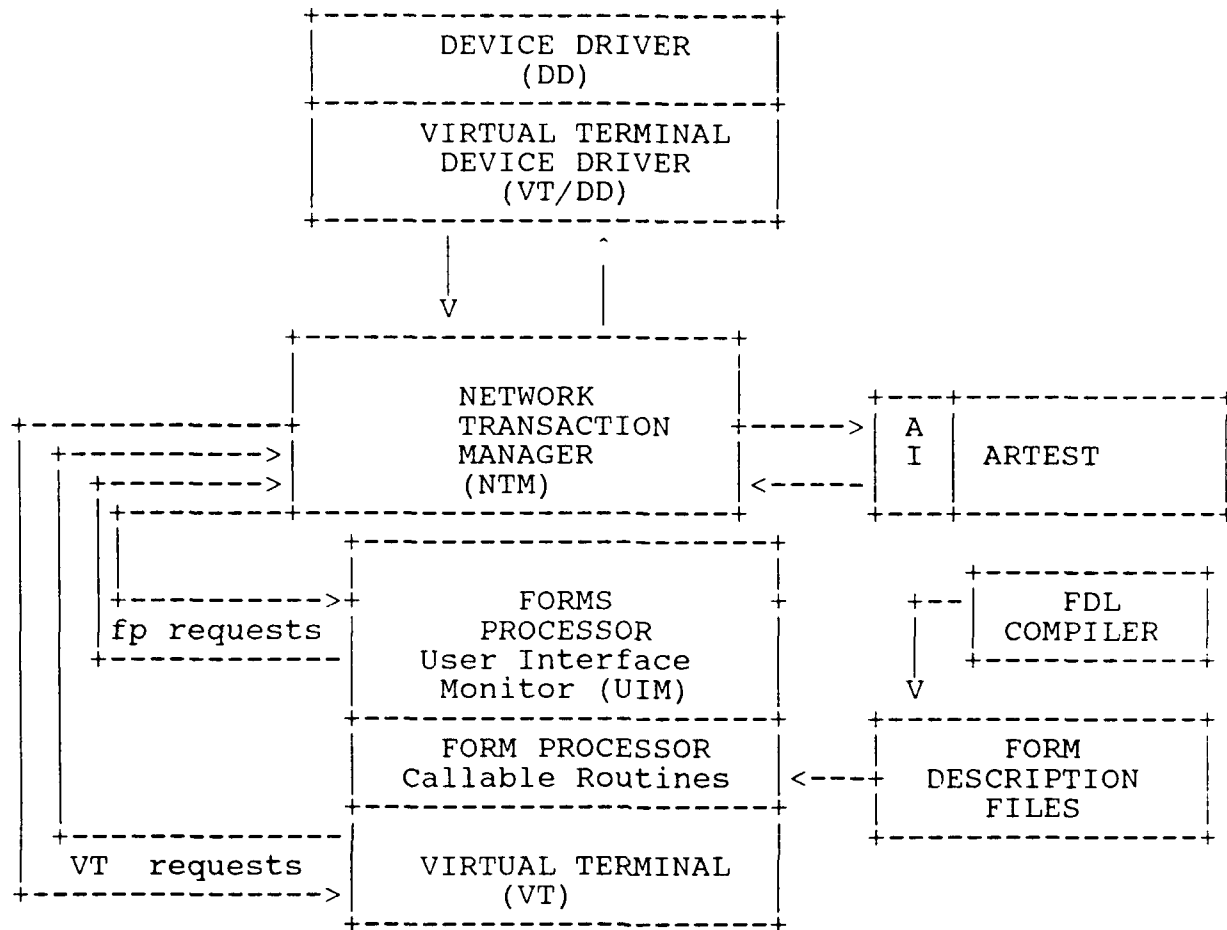


Figure 3-3 FP in IISS Environment

3.3.1 Application

The FP interface for IISS applications is the set of callable execution time routines available for an application program for form processing. These routines are defined in the IISS Form Processor User's Manual (UM 620144200B). The FP routines allow application programs and their users to communicate through predefined forms on a terminal. Again the application may directly interface with the FP through the FP routines or through the AI routines. In either case, the calling sequence is exactly the same.

3.3.2 Virtual Terminal

The FP interfaces with the Virtual Terminal (VT) by means of using the VT callable routines. Use of these routines is only necessary when initializing the FP (INITFP), terminating the FP (TERMFP), and outputting data to and receiving data from the terminal (OISCR, OUTSCR).

3.3.3 Forms Definition Language

The FP interfaces with the FDL by use of the Forms Definition File. This file contains the binary definition of the forms that the FP may use. It simply reads in a form by form name once an Open Form request is issued for a given form.

3.4 Program Interrupts

This section does not apply to the detailed design of the Form Processor.

3.5 Timing and Sequencing Description

Timing and sequencing for the control logic involved in referencing each CPC of the Form Processor is based on the sending of AI messages by application programs and the delivery of these messages through the NTM to the UIM of the FP. These messages are processed on a first come, first processed basis.

3.6 Special Control Features

The detailed design of the Form Processor does not include any special control features as defined in the ICAM Documentation Standards manual.

3.7 Storage Allocation

The Form Processor executable is 423 blocks. The FP allocates memory for form elements at run time. The amount of memory used depends upon the number of open forms and the number of fields on these forms.

3.7.1 Data Base Definition

Section 3.2.3 of the Form Processor Development Specification (DS 620144200B) describes the FP internal data structures.

3.7.1.1 File Descriptions

Form Definition files are the only external data used by the FP.

1. FILE NAME: formname.FD - Form Definition file. A complete description of the Form Definition file which is a binary file is contained in Appendix B of the Forms Language Compiler Development Specification (DS 620144401B). The name of this file is dependent upon the form it describes.

PURPOSE: This file contains information about the structure and attributes of a form that is used a run time by the FP.

DECLARATION:

```
typedef struct      /* version number record */
{
    char rectyp;      /* '1' */
    int  vernum;      /* current version number (2) */
    char linefeed;
} VERREC;

typedef struct      /* form record */
{
    char  form_name[10]; /* form name */
    char  background[10]; /* background name */
    short row;          /* starting row */
    short col;          /* starting col */
    short width;        /* width */
    short depth;        /* depth */
    short n_txtflds;    /* number of text fields */
    short n_datflds;    /* number of data fields */
    short s_txtbuf;     /* size of the text buffer */
    short s_defbuf;     /* size of the default buffer */
    char  linefeed;
} FRMREC;

typedef struct      /* text record */
{
    short row;          /* starting row */
    short col;          /* starting col */
    short len;          /* total length */
    char  linefeed;
} TXTREC;
```

```
typedef struct      /* field record */
{
    char fld_name[10]; /* field name */
    char fld_type;     /* field type (F, I, W, A) */
    short row;         /* starting row */
    short col;         /* starting col */
    short width;       /* field width */
    short depth;       /* field depth */
    int min_value;     /* minimum value (if any) */
    int max_value;     /* maximum value (if any) */
    char helpline[80]; /* help text */
    char disp_att[10]; /* display attribute */
    short n_formats;   /* number of formats */
    char format[12][2]; /* format strings */
    short n_arydefs;   /* number of dimensions */
    struct /* dimension specification */
    {
        char dir; /* repeat direction (H, V) */
        short cnt; /* actual repeat count */
        short sp; /* number of spaces between
                    repetitions */
        short dsp_size; /* display repeat count */
    } array_def[3];
    char linefeed;
} FLDREC;

typedef struct { /* run time relative positioning
info */
    POS posnod;
    NAME mynam, hnam, vnam;
} RELREC;
```

3.8 Object Code Creation

The FP routines were compiled with a C compiler developed by Interactive Software under VAX/VMS. The source is portable to other compilers on machines such as the IBM.

3.9 Adaptation Data

The FP source can be compiled using any UNIX version 7 compatible C compiler.

3.10 Detailed Design Description

3.10.1 Main Program List

The following is a list of all "Main Programs" which are modules that are not called by any other module being documented here. These modules are either program entry points or, if they are hooked into another set of programs via subroutine calls, they are the points the external programs can call and therefore enter through. To differentiate between the two types of entry points, look at the individual Module Documentation (section 3.10.8) and look at Module Type for each

of the Main Program modules listed. Note whether the routine is a Program, Subroutine, or Function. If it is a Program, it is truly a main program entry point. If not, then it is merely called by other programs not being documented here.

FORM PROCESSOR Main Program List

Module Name -----	Purpose -----
GARPOS	GET ARRAY OFFSET POSITION OF FIELD
MONITR/MAIN	MAIN MODULE FOR MONITOR/UIS/FP PROCESS
PRNDSP	PRINT DISPLAY LIST
PRNOPN	PRINT OPEN LIST
PRNUID	PRINT UID
PRNUSR	PRINT USER

3.10.2 Module List

The following is a list of all the modules being documented here along with their purpose. Each module has a unique name, no matter what language it was written in.

FORM PROCESSOR Module List

Module Name -----	Purpose -----
ACRPOS	ABSOLUTIZE CURSOR POSITION OF FIELD
ADDELM	ADD ELEMENT
ADDFRM	ADD FORM TO WINDOW
ADJSTR	ADJUST FORM PROCESSOR STRUSCTURE
CALLFP	CALL FP ROUTINES
CANITM	CANONICALIZE ITEM
CHGLDV	CHANGE LOGICAL DEVICE
CHGPRC	CHANGE PRECEDENCE OF WINDOW OR LOGICAL DEVICE
CLSFRM	CLOSE FORM
CLSLDV	CLOSE LOGICAL DEVICE
CMPFLD	COMPUTE FIELD
CMPFLD/EVAL	EVALUATE FIELD EXPRESSION
COPFLD	COPY FIELD
COPFLD/CPYFLD	INTERNAL COPY FIELD
COPFRM	COPY FORM
CURPOS	GET CURSOR POSITION
CURPOS/FNDPCP	FIND CURSOR POSITION
DBCFCNC	CHECK FUNCTION
DBCROL	CHECK ROLE
DELFLD	DELETE FIELD

FORM PROCESSOR Module List

Module Name -----	Purpose -----
DELFLD/DELEXP	DELETE EXPRESSION
ESCPY	EXTERNAL STRING COPY
FNDFLD	FIND FIELD
FNDMSG	FIND MESSAGE
FNDMSG/CODSCH	CODE SEARCH
FNDMSG/OUMSGF	OPEN USER MESSAGE FILE
FNFPWN	FIND FORM PROCESSOR WINDOW
FUISWN	FIND UIS WINDOW
GARPOS	GET ARRAY OFFSET POSITION OF FIELD
GATDEF	GET ATTRIBUTE DEFINITION
GDATA	GET DATA
GDATA/GETBUF	GET BUFFER
GDATA LN	GET DATA LENGTH
GDATA LN/GBUFLN	GET BUFFER LENGTH
GDVINP	GET DEVICE INPUT
GETATT	GET ATTRIBUTE
GETBAK	GET BACKGROUND ATTRIBUTE
GETCUR	GET CURSOR POSITION
GETCUR/CONCAT	CONCATENATE STRING TO CURRENT NAME
GOFPTR	GET OPEN FROM POINTER
GPAGE	GET PAGE

FORM PROCESSOR Module List

Module Name -----	Purpose -----
GWINDO	GET WINDOW
INITVT	INITIAL VIRTUAL TERMINAL INTERFACE
INQLDV	INQUIRE LOGICAL DEVICE
INSCR	INPUT SCREEN
MABSAT	MAP ABSOLUTE ATTRIBUTE
MAKAP	MAKE APPLICATION STRUCTURE
MAKFLD	MAKE FIELD
MAKFPD	MAKE FORM PROCESSOR DATA (LOGICAL DEVICE STRUCTURE)
MAKPD	MAKE PHYSICAL DEVICE STRUCTURE
MAKUSR	MAKE USER
MONITR/GETPD	GET PHYSICAL DEVICE
MONITR/MAIN	MAIN MODULE FOR MONITOR/UIS/FP PROCESS
OISCR/ADDCMD	ADD COMMAND TO BUFFER
OISCR/CMPALL	COMPUTE ALL CALCULATED FIELDS
OISCR/CNGMSG	CHECK FOR AND PROCESS CHANGE MESSAGE REQUESTS
OISCR/DSPSCR	DISPLAY SCREEN
OISCR/EVTBUF	EMPTY VTI BUFFER
OISCR/FVTBUF	FILL VTI BUFFER
OISCR/GATINF	GET ATTRIBUTE INFO
OISCR/PROCFLD	PROCESS FIELD

FORM PROCESSOR Module List

Module Name -----	Purpose -----
OISCR/PROCWIN	PROCESS WINDOW
OISCR/RSTINP	RESET INPUT FLAGS
OISCR/RSTMAT	RESET TEMPORAY ATTRIBUTES
OISCR/SETWIN	SET WINDOW
ONWISC	OUTPUT (NO WAIT) / INPUT SCREEN
OPNFRM	OPEN FORM
OPNFRM/BDBUFF	BUILD DEFAULT BUFFER
OPNFRM/BFLDDB	BUILD FIELD DEFAULT BUFFER
OPNFRM/BRPNOD	BUILD RELATIVE POSITION NODE
OPNFRM/BTBUFF	BUILD TEXT BUFFER
OPNFRM/PARY	PROCESS ARRAY
OPNFRM/PDREC	PROCESS FIELD RECORD
OPNFRM/PFREC	PROCESS FORM RECORD
OPNFRM/PFRM	PROCESS FORM
OPNFRM/PITM	PROCESS ITEM
OPNFRM/PTREC	PROCESS TEXT RECORD
OPNFRM/PWIN	PROCESS WINDOW
OPNLDV	OPEN LOGICAL DEVICE
OUTSCR	OUTPUT SCREEN
PARFQN	PARSE FULLY QUALIFIED NAME
PDATA	PUT FORM DATA

FORM PROCESSOR Module List

Module Name -----	Purpose -----
PDATA/PUTBUF	PUT BUFFER
PDVOTP	PUT DEVICE OUTPUT
PMSGLC	PUT MESSAGE LINE CODE
PMSGLS	PUT MESSAGE LINE STRING
POSCUR	POSITION CURSOR
POSCUR/FNFITM	FIND FIRST ITEM OF FIELD
PRNAP	PRINT APPLICATION
PRNDSP	PRINT DISPLAY LIST
PRNFLD	PRINT FIELD
PRNOPN	PRINT OPEN LIST
PRNPD	PRINT PHYSICAL DEVICE
PRNUID	PRINT UID
PRNUSR	PRINT USER
PTHPTR	GET PATH POINTER
PTHPTR/ARRAY	PROCESS ARRAY
PTHPTR/FIELD	MATCH FIELD
PTHPTR/FORM	PROCESS FORM
PTHPTR/FOUND	HAS ANYTHING BEEN FOUND?
PTHPTR/ITEM	PROCESS ITEM
PTHPTR/WINDOW	PROCESS WINDOW
PUTATT	PUT ATTRIBUTES

FORM PROCESSOR Module List

Module Name -----	Purpose -----
PUTATT/AABSAT	ATTRIBUTE ABSOLUTE SET ATTRIBUTE
PUTBAK	PUT BACKGROUND ATTRIBUTES
PUTCUR	PUT CURSOR
PUTLOC	PUT LOCATION
RMVAP	REMOVE APPLICATION
RMVFPD	REMOVE FORM PROCESSOR DATA STRUCTURE
RMVPAG	REMOVE PAGE
RMVPD	REMOVE PHYSICAL DEVICE DATA STRUCTURE
RPLFRM	REPLACE FORM
RSVATT	RESOLVE ATTRIBUTE
RSVATT/RSVRST	RESOLVE REST
RSVEXP	RESOLVE EXPRESSIONS
RSVEXP/BLDEXP	BUILD EXPRESSION TREE
SFPDAP	SET FORM PROCESSOR DATA STRUCTURE FOR APPLICATION
STUPFP	SET UP FORM PROCESSOR DATA STRUCTURES
SYSMSG	SYSTEM MESSAGE ROUTINE
TERMVT	TERMINATE VIRTUAL TERMINAL INTERFACE
TRMDRV	TERMINATE DEVICE DRIVER
TRMUSR	TERMINATE USER
UIS	USR INTERFACE SERVICES

FORM PROCESSOR Module List

Module Name -----	Purpose -----
UIS/FLWINF	FILL WINDOW INFORMATION
UIS/FLWNST	FILL WINDOW MANAGER STRUCTURE
UIS/PRCINP	PROCESS INPUT
UIS/PRCWND	PROCESS WINDOW
UIS/STRTAP	START APPLICATION
UIS/STRTPD	START PHYSICAL DEVICE
ULKFPD	UNLINKK FPD

3.10.3 External Routines List

The following is a list of all routines or functions not documented here that are called by modules that are documented here. The first caller, in alphabetical order, is listed as well. The specification in which any module is documented may be found in the Module Documentation Index (Document Number CM 620100001). See section 3.10.6 for a list of the modules that call each of these external routines.

FORM PROCESSOR External Routines List

Module Name -----	First User -----
ABORT	OPNFRM/BRPNOD
ABS	OISCR/PROCWIN
ATOI	MAKAP
BLDCMD	MONITR/MAIN
BLN	COPFLD/CPYFLD
CALLOC	MAKAP
CBIT	DELFLD
CBPTR	PDATA/PUTBUF
CFREE	MAKAP
DBCLSE	MONITR/MAIN
DBCOM	UIS/STRTAP
DBCUPR	UIS
DBGAPD	UIS/STRTAP
DBOPEN	MONITR/MAIN
DOATTR	PRNFLD
DOITEM	PRNFLD
DOWIND	PRNFLD
DSPMSG	RMVAP
ERRPRO	SYMSG
FCLOSE	RMVPD
FEOF	OPNFRM/BDBUFF
FERROR	OPNFRM/BRPNOD
FFBCA	COPFLD/CPYFLD
FOPEN	UIS/STRTPD
FPRINTF	MONITR/MAIN
FREAD	OPNFRM/PTREC
FREE	GETCUR
FREMSG	RMVFPD
FSEARCH	UIS/STRTPD
FSEEK	MONITR/MAIN
FTELL	GDVINP
FWRITE	PDVOTP
GETC	OPNFRM/BTBUFF
GETW	MONITR/MAIN
GVTICMD	MONITR/MAIN
GVTINW	CALLFP
INITAL	MONITR/MAIN
ISALNUM	PTHPTR
ISCNTRL	OPNFRM/BTBUFF
ISDIGIT	PTHPTR
ISEND	UIS/STRTAP

FORM PROCESSOR External Routines List

Module Name -----	First User -----
ISPRINT	PMSGLS
LOCALTIME	OISCR/DSPSCR
MALLOC	OPNFRM/PITM
MATOI	UIS/PRCINP
MAX	RMVPAG
MEMCMP	UIS/FLWNST
MEMCPY	PARFQN
MEMSET	GDVINP
MIN	CALLFP
NSEND	MONITR/MAIN
OBIND	DBCROL
ODFINN	DBCFNC
OEXEC	DBCROL
OFETCH	DBCROL
OSQL3	DBCFNC
PBPTR	COPFLD/CPYFLD
PFINP	INSCR
PRINTF	PRNUSR
PUTC	MONITR/MAIN
PUTW	MONITR/MAIN
RCV	MONITR/MAIN
REWIND	FNDMSG/CODSCH
SBIT	MAKAP
SIGABT	TRMUSR
SNDVTI	CALLFP
SPRINTF	OPNFRM
STRASN	OPNFRM/PWIN
STRCAT	OISCR/PROCFLD
STRCHR	PARFQN
STRCMP	MONITR/GETPD
STRCPY	MAKFLD
STRLEN	OISCR/FVTBUF
STRNCMP	FNDMSG
STRNCPY	ESCPY
STRNLOC	CANITM
STRNUPC	CANITM
STRRCHR	PARFQN
STRUPC	COPFRM
TIME	OISCR/DSPSCR
TOUPPER	PTHPTR
TRMNAT	MONITR/MAIN

3.10.4 Include File List

The following is a list of all include files called in by modules being documented here. Each include file has a unique name regardless of the language being used. The purpose of each include file is listed as well. A more complete description of each include file is given in section 3.10.9. The purpose listed is the one that is in the source code of the include file.

A purpose of "**** PURPOSE NOT FOUND BY STRIPPER ****" indicates that a purpose statement was not written into the include file itself. The most common reason for this is that the include file comes from system libraries that were not developed by the project, such as 'C' libraries that are provided with the 'C' compiler.

See section 3.10.6 for a set of lists which show all the modules which call in each of these include files.

FORM PROCESSOR Include File List

File Name -----	Purpose -----
BITS	INCLUDE FILE FOR BIT MANIPULATION ROUTINES
CICODE	Command Interpreter CODEs
CTLCHR	CONTROL CHARACTERS
CTYPE	**** PURPOSE NOT FOUND BY STRIPPER ****
CURSORI	CURSOR description
DBASEI	DATABASE ITERFACE
FFV2	FORM FILE FORMAT - VERSION 2
FPCODE	FORM PROCESSOR RETURN CODES
FPD	FORM PROCESSOR DATA
FPDINI	FPD INITIALIZATION
FPEMSG	FORM PROCESSOR ERROR MESSAGES
FPPARM	FORM PROCESSOR PARAMETERS
FUNCTS	FUNCTION DEFINITIONS
NTM	NTM INTERFACE INCLUDE FILE
ORACLE	data delcarations for programs that access ORACLE
ORCODE	ORacle CODEs
STDIO	**** PURPOSE NOT FOUND BY STRIPPER ****
STDTyp	STANDARD TYPE DEFINITIONS
TIME	**** PURPOSE NOT FOUND BY STRIPPER ****
UISFM	UIS FORM
VTICOM	VTI COMMUNICATION DEFINITIONS

3.10.5 Where Include File Used List

The following lists each include file from 3.10.4 and all the modules documented in this specification which include them. The purpose of each module is listed as well.

FORM PROCESSOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
--------------------------	-------------------------	----------------------------

BITS

COPFLD	COPY FIELD
COPFLD/CP	INTERNAL COPY FIELD
DELFLD	DELETE FIELD
DELFLD/DE	DELETE EXPRESSION

CICODE

DBCFCNC	CHECK FUNCTION
DBCROL	CHECK ROLE

CTLCHR

GDVINP	GET DEVICE INPUT
INSCR	INPUT SCREEN
MAKUSR	MAKE USER
MONITR/GE	GET PHYSICAL DEVICE
MONITR/MA	MAIN MODULE FOR MONITOR/UIS/FP PROCESS
OISCR/ADD	ADD COMMAND TO BUFFER
OISCR/CMP	COMPUTE ALL CALCULATED FIELDS
OISCR/CNG	CHECK FOR AND PROCESS CHANGE MESSAGE REQUESTS
OISCR/DSP	DISPLAY SCREEN
OISCR/EVT	EMPTY VTI BUFFER
OISCR/FVT	FILL VTI BUFFER
OISCR/GAT	GET ATTRIBUTE INFO
OISCR/PRO	PROCESS FIELD
OISCR/PRO	PROCESS WINDOW
OISCR/RST	RESET INPUT FLAGS
OISCR/RST	RESET TEMPORAY ATTRIBUTES
OISCR/SET	SET WINDOW
ONWISC	OUTPUT (NO WAIT) / INPUT SCREEN
OUTSCR	OUTPUT SCREEN
RMVFPD	REMOVE FORM PROCESSOR DATA STRUCTURE
UIS	USR INTERFACE SERVICES
UIS/FLWIN	FILL WINDOW INFORMATION

FORM PROCESSOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	UIS/FLWNS	FILL WINDOW MANAGER STRUCTURE
	UIS/PRCIN	PROCESS INPUT
	UIS/PRCWN	PROCESS WINDOW
	UIS/STRTA	START APPLICATION
	UIS/STRTP	START PHYSICAL DEVICE

CTYPE

CMPFLD	COMPUTE FIELD
CMPFLD/EV	EVALUATE FIELD EXPRESSION
INSCR	INPUT SCREEN
OISCR/ADD	ADD COMMAND TO BUFFER
OISCR/CMP	COMPUTE ALL CALCULATED FIELDS
OISCR/CNG	CHECK FOR AND PROCESS CHANGE MESSAGE REQUESTS
OISCR/DSP	DISPLAY SCREEN
OISCR/EMT	EMPTY VTI BUFFER
OISCR/FVT	FILL VTI BUFFER
OISCR/GAT	GET ATTRIBUTE INFO
OISCR/PRO	PROCESS FIELD
OISCR/PRO	PROCESS WINDOW
OISCR/RST	RESET INPUT FLAGS
OISCR/RST	RESET TEMPORAY ATTRIBUTES
OISCR/SET	SET WINDOW
ONWISC	OUTPUT (NO WAIT) / INPUT SCREEN
OPNFRM	OPEN FORM
OPNFRM/BD	BUILD DEFAULT BUFFER
OPNFRM/BF	BUILD FIELD DEFAULT BUFFER
OPNFRM/BR	BUILD RELATIVE POSITION NODE
OPNFRM/BT	BUILD TEXT BUFFER
OPNFRM/PA	PROCESS ARRAY
OPNFRM/PD	PROCESS FIELD RECORD
OPNFRM/PF	PROCESS FORM RECORD
OPNFRM/PF	PROCESS FORM
OPNFRM/PI	PROCESS ITEM
OPNFRM/PT	PROCESS TEXT RECORD
OPNFRM/PW	PROCESS WINDOW
OUTSCR	OUTPUT SCREEN

FORM PROCESSOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	PDATA	PUT FORM DATA
	PDATA/PUT	PUT BUFFER
	PMSGLS	PUT MESSAGE LINE STRING
	PTHPTR	GET PATH POINTER
	PTHPTR/AR	PROCESS ARRAY
	PTHPTR/FI	MATCH FIELD
	PTHPTR/FO	PROCESS FORM
	PTHPTR/FO	HAS ANYTHING BEEN FOUND?
	PTHPTR/IT	PROCESS ITEM
	PTHPTR/WI	PROCESS WINDOW
	RSVEXP	RESOLVE EXPRESSIONS
	RSVEXP/BL	BUILD EXPRESSION TREE
CURSORI	DBCFCNC	CHECK FUNCTION
	DBCROL	CHECK ROLE
DBASEI	MONITR/GE	GET PHYSICAL DEVICE
	MONITR/MA	MAIN MODULE FOR MONITOR/UIS/FP PROCESS
	UIS	USR INTERFACE SERVICES
	UIS/FLWIN	FILL WINDOW INFORMATION
	UIS/FLWNS	FILL WINDOW MANAGER STRUCTURE
	UIS/PRCIN	PRCESS INPUT
	UIS/PRCWN	PRCESS WINDOW
	UIS/STRTA	START APPLICATION
	UIS/STRTP	START PHYSICAL DEVICE
FFV2	OPNFRM	OPEN FORM
	OPNFRM/BD	BUILD DEFAULT BUFFER

FORM PROCESSOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	OPNFRM/BF	BUILD FIELD DEFAULT BUFFER
	OPNFRM/BR	BUILD RELATIVE POSITION NODE
	OPNFRM/BT	BUILD TEXT BUFFER
	OPNFRM/PA	PROCESS ARRAY
	OPNFRM/PD	PROCESS FIELD RECORD
	OPNFRM/PF	PROCESS FORM RECORD
	OPNFRM/PF	PROCESS FORM
	OPNFRM/PI	PROCESS ITEM
	OPNFRM/PT	PROCESS TEXT RECORD
	OPNFRM/PW	PROCESS WINDOW

FPCODE

ADDELM	ADD ELEMENT
ADDFRM	ADD FORM TO WINDOW
CALLFP	CALL FP ROUTINES
CHGLDV	CHANGE LOGICAL DEVICE
CLSFRM	CLOSE FORM
CLSLDV	CLOSE LOGICAL DEVICE
CMPFLD	COMPUTE FIELD
CMPFLD/EV	EVALUATE FIELD EXPRESSION
COPFLD	COPY FIELD
COPFLD/CP	INTERNAL COPY FIELD
COPFRM	COPY FORM
DBCFCNC	CHECK FUNCTION
DBCROL	CHECK ROLE
DELFLD	DELETE FIELD
DELFLD/DE	DELETE EXPRESSION
FNDMSG	FIND MESSAGE
FNDMSG/CO	CODE SEARCH
FNDMSG/OU	OPEN USER MESSAGE FILE
GATDEF	GET ATTRIBUTE DEFINITION
GDATA	GET DATA
GDATA/GET	GET BUFFER
GDATLN	GET DATA LENGTH
GDATLN/GB	GET BUFFER LENGTH
GDVINP	GET DEVICE INPUT
GETATT	GET ATTRIBUTE

FORM PROCESSOR Where-include -file-used List

Include File -----	Module Name -----	Module Purpose -----
	GETBAK	GET BACKGROUND ATTRIBUTE
	GETCUR	GET CURSOR POSITION
	GETCUR/CO	CONCATENATE STRING TO CURRENT NAME
	GPAGE	GET PAGE
	GWINDO	GET WINDOW
	INITVT	INITIAL VIRTUAL TERMINAL INTERFACE
	INQLDV	INQUIRE LOGICAL DEVICE
	INSCR	INPUT SCREEN
	MAKAP	MAKE APPLICATION STRUTURE
	MAKFLD	MAKE FIELD
	MAKPD	MAKE PHYSICAL DEVICE STRUCTURE
	MAKUSR	MAKE USER
	MONITR/GE	GET PHYSICAL DEVICE
	MONITR/MA	MAIN MODULE FOR MONITOR/UIS/FP PROCESS
	OISCR/ADD	ADD COMMAND TO BUFFER
	OISCR/CMP	COMPUTE ALL CALCULATED FIELDS
	OISCR/CNG	CHECK FOR AND PROCESS CHANGE MESSAGE REQUESTS
	OISCR/DSP	DISPLAY SCREEN
	OISCR/EVT	EMPTY VTI BUFFER
	OISCR/FVT	FILL VTI BUFFER
	OISCR/GAT	GET ATTRIBUTE INFO
	OISCR/PRO	PROCESS FIELD
	OISCR/PRO	PROCESS WINDOW
	OISCR/RST	RESET INPUT FLAGS
	OISCR/RST	RESET TEMPORAY ATTRIBUTES
	OISCR/SET	SET WINDOW
	ONWISC	OUTPUT (NO WAIT) / INPUT SCREEN
	OPNFRM	OPEN FORM
	OPNFRM/BD	BUILD DEFAULT BUFFER
	OPNFRM/BF	BUILD FIELD DEFAULT BUFFER
	OPNFRM/BR	BUILD RELATIVE POSITION NODE
	OPNFRM/BT	BUILD TEXT BUFFER
	OPNFRM/PA	PROCESS ARRAY
	OPNFRM/PD	PROCESS FIELD RECORD
	OPNFRM/PF	PROCESS FORM RECORD
	OPNFRM/PF	PROCESS FORM
	OPNFRM/PI	PROCESS ITEM
	OPNFRM/PT	PROCESS TEXT RECORD
	OPNFRM/PW	PROCESS WINDOW

FORM PROCESSOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	OPNLDV	OPEN LOGICAL DEVICE
	OUTSCR	OUTPUT SCREEN
	PARFQN	PARSE FULLY QUALIFIED NAME
	PDATA	PUT FORM DATA
	PDATA/PUT	PUT BUFFER
	PDVOTP	PUT DEVICE OUTPUT
	PMSGLS	PUT MESSAGE LINE STRING
	PTHPTR	GET PATH POINTER
	PTHPTR/AR	PROCESS ARRAY
	PTHPTR/FI	MATCH FIELD
	PTHPTR/FO	PROCESS FORM
	PTHPTR/FO	HAS ANYTHING BEEN FOUND?
	PTHPTR/IT	PROCESS ITEM
	PTHPTR/WI	PROCESS WINDOW
	PUTATT	PUT ATTRIBUTES
	PUTATT/AA	ATTRIBUTE ABSOLUTE SET ATTRIBUTE
	PUTBAK	PUT BACKGROUND ATTRIBUTES
	PUTCUR	PUT CURSOR
	PUTLOC	PUT LOCATION
	RMVPAG	REMOVE PAGE
	RMVPD	REMOVE PHYSICAL DEVICE DATA STRUCTURE
	RPLFRM	REPLACE FORM
	RSVEXP	RESOLVE EXPRESSIONS
	RSVEXP/BL	BUILD EXPRESSION TREE
	SFPDAP	SET FORM PROCESSOR DATA STRUCTURE FOR APPLICATION
	STUPFP	SET UP FORM PROCESSOR DATA STRUCTURES
	SYMSG	SYSTEM MESSAGE ROUTINE
	TERMTV	TERMINATE VIRTUAL TERMINAL INTERFACE
	TRMDRV	TERMINATE DEVICE DRIVER
	TRMUSR	TERMINATE USER
	UIS	USR INTERFACE SERVICES
	UIS/FLWIN	FILL WINDOW INFORMATION
	UIS/FLWNS	FILL WINDOW MANAGER STRUCTURE
	UIS/PRCIN	PROCESS INPUT
	UIS/PRCWN	PROCESS WINDOW
	UIS/STRTA	START APPLICATION
	UIS/STRTP	START PHYSICAL DEVICE

FORM PROCESSOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
--------------------------	-------------------------	----------------------------

FPD

ACRPOS	ABSOLUTIZE CURSOR POSITION OF FIELD
ADDELM	ADD ELEMENT
ADDFRM	ADD FORM TO WINDOW
ADJSTR	ADJUST FORM PROCESSOR STRUSCTURE
CALLFP	CALL FP ROUTINES
CANITM	CANONICALIZE ITEM
CHGLDV	CHANGE LOGICAL DEVICE
CHGPCR	CHANGE PRECEDENCE OF WINDOW OR LOGICAL DEVICE
CLSFRM	CLOSE FORM
CLSLDV	CLOSE LOGICAL DEVICE
CMPFLD	COMPUTE FIELD
CMPFLD/EV	EVALUATE FIELD EXPRESSION
COPFLD	COPY FIELD
COPFLD/CP	INTERNAL COPY FIELD
COPFRM	COPY FORM
CURPOS	GET CURSOR POSITION
CURPOS/FN	FIND CURSOR POSITION
DEFLD	DELETE FIELD
DEFLD/DE	DELETE EXPRESSION
FNDFLD	FIND FIELD
FNDMSG	FIND MESSAGE
FNDMSG/CO	CODE SEARCH
FNDMSG/OU	OPEN USER MESSAGE FILE
FNFPWN	FIND FORM PROCESSOR WINDOW
FUISWN	FIND UIS WINDOW
GARPOS	GET ARRAY OFFSET POSITION OF FIELD
GATDEF	GET ATTRIBUTE DEFINITION
GDATA	GET DATA
GDATA/GET	GET BUFFER
GDATLN	GET DATA LENGTH
GDATLN/GB	GET BUFFER LENGTH
GDVINP	GET DEVICE INPUT
GETATT	GET ATTRIBUTE
GETBAK	GET BACKGROUND ATTRIBUTE

FORM PROCESSOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	GETCUR	GET CURSOR POSITION
	GETCUR/CO	CONCATENATE STRING TO CURRENT NAME
	GOFPTR	GET OPEN FROM POINTER
	GPAGE	GET PAGE
	GWINDO	GET WINDOW
	INITVT	INITIAL VIRTUAL TERMINAL INTERFACE
	INQLDV	INQUIRE LOGICAL DEVICE
	INSCR	INPUT SCREEN
	MABSAT	MAP ABSOLUTE ATTRIBUTE
	MAKAP	MAKE APPLICATION STRUCTURE
	MAKFLD	MAKE FIELD
	MAKFPD	MAKE FORM PROCESSOR DATA (LOGICAL DEVICE STRUCTURE)
	MAKPD	MAKE PHYSICAL DEVICE STRUCTURE
	MAKUSR	MAKE USER
	MONITR/GE	GET PHYSICAL DEVICE
	MONITR/MA	MAIN MODULE FOR MONITOR/UIS/FP PROCESS
	OISCR/ADD	ADD COMMAND TO BUFFER
	OISCR/CMP	COMPUTE ALL CALCULATED FIELDS
	OISCR/CNG	CHECK FOR AND PROCESS CHANGE MESSAGE REQUESTS
	OISCR/DSP	DISPLAY SCREEN
	OISCR/EVT	EMPTY VTI BUFFER
	OISCR/FVT	FILL VTI BUFFER
	OISCR/GAT	GET ATTRIBUTE INFO
	OISCR/PRO	PROCESS FIELD
	OISCR/PRO	PROCESS WINDOW
	OISCR/RST	RESET INPUT FLAGS
	OISCR/RST	RESET TEMPORAY ATTRIBUTES
	OISCR/SET	SET WINDOW
	ONWISC	OUTPUT (NO WAIT) / INPUT SCREEN
	OPNFRM	OPEN FORM
	OPNFRM/BD	BUILD DEFAULT BUFFER
	OPNFRM/BF	BUILD FIELD DEFAULT BUFFER
	OPNFRM/BR	BUILD RELATIVE POSITION NODE
	OPNFRM/BT	BUILD TEXT BUFFER
	OPNFRM/PA	PROCESS ARRAY
	OPNFRM/PD	PROCESS FIELD RECORD
	OPNFRM/PF	PROCESS FORM RECORD
	OPNFRM/PF	PROCESS FORM

FORM PROCESSOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	OPNFRM/FI	PROCESS ITEM
	OPNFRM/PT	PROCESS TEXT RECORD
	OPNFRM/PW	PROCESS WINDOW
	OPNLDV	OPEN LOGICAL DEVICE
	OUTSCR	OUTPUT SCREEN
	PARFQN	PARSE FULLY QUALIFIED NAME
	PDATA	PUT FORM DATA
	PDATA/PUT	PUT BUFFER
	PDVOTP	PUT DEVICE OUTPUT
	PMSGLC	PUT MESSAGE LINE CODE
	PMSGLS	PUT MESSAGE LINE STRING
	POSCUR	POSITION CURSOR
	POSCUR/FN	FIND FIRST ITEM OF FIELD
	PRNAP	PRINT APPLICATION
	PRNDSP	PRINT DISPLAY LIST
	PRNFLD	PRINT FIELD
	PRNOPN	PRINT OPEN LIST
	PRNPD	PRINT PHYSICAL DEVICE
	PRNUID	PRINT UID
	PRNUSR	PRINT USER
	PTHPTR	GET PATH POINTER
	PTHPTR/AR	PROCESS ARRAY
	PTHPTR/FI	MATCH FIELD
	PTHPTR/FO	PROCESS FORM
	PTHPTR/FO	HAS ANYTHING BEEN FOUND?
	PTHPTR/IT	PROCESS ITEM
	PTHPTR/WI	PROCESS WINDOW
	PUTATT	PUT ATTRIBUTES
	PUTATT/AA	ATTRIBUTE ABSOLUTE SET ATTRIBUTE
	PUTBAK	PUT BACKGROUND ATTRIBUTES
	PUTCUR	PUT CURSOR
	PUTLOC	PUT LOCATION
	RMVAP	REMOVE APPLICATION
	RMVFPD	REMOVE FORM PROCESSOR DATA STRUCTURE
	RMVPAG	REMOVE PAGE
	RMVPD	REMOVE PHYSICAL DEVICE DATA STRUCTURE
	RPLFRM	REPLACE FORM
	RSVATT	RESOLVE ATTRIBUTE
	RSVATT/RS	RESOLVE REST
	RSVEXP	RESOLVE EXPRESSIONS

FORM PROCESSOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
--------------------------	-------------------------	----------------------------

RSVEXP/BL	BUILD EXPRESSION TREE
SFPDAP	SET FORM PROCESSOR DATA STRUCTURE FOR APPLICATION
STUPFP	SET UP FORM PROCESSOR DATA STRUCTURES
SYSMSG	SYSTEM MESSAGE ROUTINE
TERMVT	TERMINATE VIRTUAL TERMINAL INTERFACE
TRMDRV	TERMINATE DEVICE DRIVER
TRMUSR	TERMINATE USER
UIS	USR INTERFACE SERVICES
UIS/FLWIN	FILL WINDOW INFORMATION
UIS/FLWNS	FILL WINDOW MANAGER STRUCTURE
UIS/PRCIN	PROCESS INPUT
UIS/PRCWN	PROCESS WINDOW
UIS/STRTA	START APPLICATION
UIS/STRTP	START PHYSICAL DEVICE
ULKFPD	UNLINKK FPD

FPDINI

MONITR/GE	GET PHYSICAL DEVICE
MONITR/MA	MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FPMSG

FNDMSG	FIND MESSAGE
FNDMSG/CO	CODE SEARCH
FNDMSG/OU	OPEN USER MESSAGE FILE

FPPARM

GDATA	GET DATA
GDATA/GET	GET BUFFER
GDATA/LN	GET DATA LENGTH
GDATA/LN/GB	GET BUFFER LENGTH

FORM PROCESSOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	GETATT	GET ATTRIBUTE
	GETBAK	GET BACKGROUND ATTRIBUTE
	GETCUR	GET CURSOR POSITION
	GETCUR/CO	CONCATENATE STRING TO CURRENT NAME
	GPAGE	GET PAGE
	GWINDO	GET WINDOW
	INSCR	INPUT SCREEN
	MONITR/GE	GET PHYSICAL DEVICE
	MONITR/MA	MAIN MODULE FOR MONITOR/UIS/FP PROCESS
	OISCR/ADD	ADD COMMAND TO BUFFER
	OISCR/CMP	COMPUTE ALL CALCULATED FIELDS
	OISCR/CNG	CHECK FOR AND PROCESS CHANGE MESSAGE REQUESTS
	OISCR/DSP	DISPLAY SCREEN
	OISCR/EVT	EMPTY VTI BUFFER
	OISCR/FVT	FILL VTI BUFFER
	OISCR/GAT	GET ATTRIBUTE INFO
	OISCR/PRO	PROCESS FIELD
	OISCR/PRO	PROCESS WINDOW
	OISCR/RST	RESET INPUT FLAGS
	OISCR/RST	RESET TEMPORAY ATTRIBUTES
	OISCR/SET	SET WINDOW
	ONWISC	OUTPUT (NO WAIT) / INPUT SCREEN
	OUTSCR	OUTPUT SCREEN
	PUTATT	PUT ATTRIBUTES
	PUTATT/AA	ATTRIBUTE ABSOLUTE SET ATTRIBUTE
	PUTBAK	PUT BACKGROUND ATTRIBUTES
	RMVPAG	REMOVE PAGE
	UIS	USR INTERFACE SERVICES
	UIS/FLWIN	FILL WINDOW INFORMATION
	UIS/FLWNS	FILL WINDOW MANAGER STRUCTURE
	UIS/PRCIN	PROCESS INPUT
	UIS/PRCWN	PROCESS WINDOW
	UIS/STRTA	START APPLICATION
	UIS/STRTP	START PHYSICAL DEVICE

FUNCTS

FORM PROCESSOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	GDVINP	GET DEVICE INPUT
	INSCR	INPUT SCREEN
	MAKUSR	MAKE USER
	MONITR/GE	GET PHYSICAL DEVICE
	MONITR/MA	MAIN MODULE FOR MONITOR/UIS/FP PROCESS
	OISCR/ADD	ADD COMMAND TO BUFFER
	OISCR/CMP	COMPUTE ALL CALCULATED FIELDS
	OISCR/CNG	CHECK FOR AND PROCESS CHANGE MESSAGE REQUESTS
	OISCR/DSP	DISPLAY SCREEN
	OISCR/EVT	EMPTY VTI BUFFER
	OISCR/FVT	FILL VTI BUFFER
	OISCR/GAT	GET ATTRIBUTE INFO
	OISCR/PRO	PROCESS FIELD
	OISCR/PRO	PROCESS WINDOW
	OISCR/RST	RESET INPUT FLAGS
	OISCR/RST	RESET TEMPORAY ATTRIBUTES
	OISCR/SET	SET WINDOW
	ONWISC	OUTPUT (NO WAIT) / INPUT SCREEN
	OUTSCR	OUTPUT SCREEN

NTM

CALLFP	CALL FP ROUTINES
MONITR/GE	GET PHYSICAL DEVICE
MONITR/MA	MAIN MODULE FOR MONITOR/UIS/FP PROCESS
PDVOTP	PUT DEVICE OUTPUT
TRMDRV	TERMINATE DEVICE DRIVER
TRMUSR	TERMINATE USER
UIS	USR INTERFACE SERVICES
UIS/FLWIN	FILL WINDOW INFORMATION
UIS/FLWNS	FILL WINDOW MANAGER STRUCTURE
UIS/PRCIN	PROCESS INPUT
UIS/PRCWN	PROCESS WINDOW
UIS/STRTA	START APPLICATION
UIS/STRTP	START PHYSICAL DEVICE

FORM PROCESSOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
--------------------------	-------------------------	----------------------------

ORACLE

DBCFNC	CHECK FUNCTION
DBCROL	CHECK ROLE

ORCODE

DBCFNC	CHECK FUNCTION
DBCROL	CHECK ROLE

STDIO

FNDMSG	FIND MESSAGE
FNDMSG/CO	CODE SEARCH
FNDMSG/OU	OPEN USER MESSAGE FILE
GDVINP	GET DEVICE INPUT
INSCR	INPUT SCREEN
MONITR/GE	GET PHYSICAL DEVICE
MONITR/MA	MAIN MODULE FOR MONITOR/UIS/FP PROCESS
OISCR/ADD	ADD COMMAND TO BUFFER
OISCR/CMP	COMPUTE ALL CALCULATED FIELDS
OISCR/CNG	CHECK FOR AND PROCESS CHANGE MESSAGE REQUESTS
OISCR/DSP	DISPLAY SCREEN
OISCR/EVT	EMPTY VTI BUFFER
OISCR/FVT	FILL VTI BUFFER
OISCR/GAT	GET ATTRIBUTE INFO
OISCR/PRO	PROCESS FIELD
OISCR/PRO	PROCESS WINDOW
OISCR/RST	RESET INPUT FLAGS
OISCR/RST	RESET TEMPORAY ATTRIBUTES
OISCR/SET	SET WINDOW
ONWISC	OUTPUT (NO WAIT) / INPUT SCREEN

FORM PROCESSOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	OPNFRM	OPEN FORM
	OPNFRM/BD	BUILD DEFAULT BUFFER
	OPNFRM/BF	BUILD FIELD DEFAULT BUFFER
	OPNFRM/BR	BUILD RELATIVE POSITION NODE
	OPNFRM/BT	BUILD TEXT BUFFER
	OPNFRM/PA	PROCESS ARRAY
	OPNFRM/PD	PROCESS FIELD RECORD
	OPNFRM/PF	PROCESS FORM RECORD
	OPNFRM/PF	PROCESS FORM
	OPNFRM/PI	PROCESS ITEM
	OPNFRM/PT	PROCESS TEXT RECORD
	OPNFRM/PW	PROCESS WINDOW
	OUTSCR	OUTPUT SCREEN
	PRNAP	PRINT APPLICATION
	PRNDSP	PRINT DISPLAY LIST
	PRNFLD	PRINT FIELD
	PRNOPN	PRINT OPEN LIST
	PRNPD	PRINT PHYSICAL DEVICE
	PRNUID	PRINT UID
	PRNUSR	PRINT USER

STDTPP

ACRPOS	ABSOLUTIZE CURSOR POSITION OF FIELD
ADDELM	ADD ELEMENT
ADDFRM	ADD FORM TO WINDOW
ADJSTR	ADJUST FORM PROCESSOR STRUCTURE
CALLFP	CALL FP ROUTINES
CANITM	CANONICALIZE ITEM
CHGLDV	CHANGE LOGICAL DEVICE
CHGPCR	CHANGE PRECEDENCE OF WINDOW OR LOGICAL DEVICE
CLSFRM	CLOSE FORM
CLSLDV	CLOSE LOGICAL DEVICE
CMPFLD	COMPUTE FIELD
CMPFLD/EV	EVALUATE FIELD EXPRESSION
COPFLD	COPY FIELD
COPFLD/CP	INTERNAL COPY FIELD

FORM PROCESSOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	COPFRM	COPY FORM
	CURPOS	GET CURSOR POSITION
	CURPOS/FN	FIND CURSOR POSITION
	DELFLD	DELETE FIELD
	DELFLD/DE	DELETE EXPRESSION
	ESCPY	EXTERNAL STRING COPY
	FNDFLD	FIND FIELD
	FNDMSG	FIND MESSAGE
	FNDMSG/CO	CODE SEARCH
	FNDMSG/OU	OPEN USER MESSAGE FILE
	FNFPWN	FIND FORM PROCESSOR WINDOW
	FUISWN	FIND UIS WINDOW
	GARPOS	GET ARRAY OFFSET POSITION OF FIELD
	GATDEF	GET ATTRIBUTE DEFINITION
	GDATA	GET DATA
	GDATA/GET	GET BUFFER
	GDATLN	GET DATA LENGTH
	GDATLN/GB	GET BUFFER LENGTH
	GDVINP	GET DEVICE INPUT
	GETATT	GET ATTRIBUTE
	GETBAK	GET BACKGROUND ATTRIBUTE
	GETCUR	GET CURSOR POSITION
	GETCUR/CO	CONCATENATE STRING TO CURRENT NAME
	GOFPTR	GET OPEN FROM POINTER
	GPAGE	GET PAGE
	GWINDO	GET WINDOW
	INITVT	INITIAL VIRTUAL TERMINAL INTERFACE
	INQLDV	INQUIRE LOGICAL DEVICE
	INSCR	INPUT SCREEN
	MABSAT	MAP ABSOLUTE ATTRIBUTE
	MAKAP	MAKE APPLICATION STRUCTURE
	MAKFLD	MAKE FIELD
	MAKFPD	MAKE FORM PROCESSOR DATA (LOGICAL DEVICE STRUCTURE)
	MAKPD	MAKE PHYSICAL DEVICE STRUCTURE
	MAKUSR	MAKE USER
	MONITR/GE	GET PHYSICAL DEVICE
	MONITR/MA	MAIN MODULE FOR MONITOR/UIS/FP PROCESS
	OISCR/ADD	ADD COMMAND TO BUFFER
	OISCR/CMP	COMPUTE ALL CALCULATED FIELDS

FORM PROCESSOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	OISCR/CNG	CHECK FOR AND PROCESS CHANGE MESSAGE REQUESTS
	OISCR/DSP	DISPLAY SCREEN
	OISCR/EVT	EMPTY VTI BUFFER
	OISCR/FVT	FILL VTI BUFFER
	OISCR/GAT	GET ATTRIBUTE INFO
	OISCR/PRO	PROCESS FIELD
	OISCR/PRO	PROCESS WINDOW
	OISCR/RST	RESET INPUT FLAGS
	OISCR/RST	RESET TEMPORAY ATTRIBUTES
	OISCR/SET	SET WINDOW
	ONWISC	OUTPUT (NO WAIT) / INPUT SCREEN
	OPNFRM	OPEN FORM
	OPNFRM/BD	BUILD DEFAULT BUFFER
	OPNFRM/BF	BUILD FIELD DEFAULT BUFFER
	OPNFRM/BR	BUILD RELATIVE POSITION NODE
	OPNFRM/BT	BUILD TEXT BUFFER
	OPNFRM/PA	PROCESS ARRAY
	OPNFRM/PD	PROCESS FIELD RECORD
	OPNFRM/PF	PROCESS FORM RECORD
	OPNFRM/PF	PROCESS FORM
	OPNFRM/PI	PROCESS ITEM
	OPNFRM/PT	PROCESS TEXT RECORD
	OPNFRM/PW	PROCESS WINDOW
	OPNLDV	OPEN LOGICAL DEVICE
	OUTSCR	OUTPUT SCREEN
	PARFQN	PARSE FULLY QUALIFIED NAME
	PDATA	PUT FORM DATA
	PDATA/PUT	PUT BUFFER
	PDVOTP	PUT DEVICE OUTPUT
	PMSGLC	PUT MESSAGE LINE CODE
	PMSGLS	PUT MESSAGE LINE STRING
	POSCUR	POSITION CURSOR
	POSCUR/FN	FIND FIRST ITEM OF FIELD
	PRNAP	PRINT APLICATION
	PRNDSP	PRINT DISPLAY LIST
	PRNFLD	PRINT FIELD
	PRNOPN	PRINT OPEN LIST
	PRNPD	PRINT PHYSICAL DEVICE
	PRNUID	PRINT UID

FORM PROCESSOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	PRNUSR	PRINT USER
	PTHPTR	GET PATH POINTER
	PTHPTR/AR	PROCESS ARRAY
	PTHPTR/FI	MATCH FIELD
	PTHPTR/FO	PROCESS FORM
	PTHPTR/FO	HAS ANYTHING BEEN FOUND?
	PTHPTR/IT	PROCESS ITEM
	PTHPTR/WI	PROCESS WINDOW
	PUTATT	PUT ATTRIBUTES
	PUTATT/AA	ATTRIBUTE ABSOLUTE SET ATTRIBUTE
	PUTBAK	PUT BACKGROUND ATTRIBUTES
	PUTCUR	PUT CURSOR
	PUTLOC	PUT LOCATION
	RMVAP	REMOVE APPLICATION
	RMVFPD	REMOVE FORM PROCESSOR DATA STRUCTURE
	RMVPAG	REMOVE PAGE
	RMVPD	REMOVE PHYSICAL DEVICE DATA STRUCTURE
	RPLFRM	REPLACE FORM
	RSVATT	RESOLVE ATTRIBUTE
	RSVATT/RS	RESOLVE REST
	RSVEXP	RESOLVE EXPRESSIONS
	RSVEXP/BL	BUILD EXPRESSION TREE
	SFPDAP	SET FORM PROCESSOR DATA STRUCTURE FOR APPLICATION
	STUPFP	SET UP FORM PROCESSOR DATA STRUCTURES
	SYMSG	SYSTEM MESSAGE ROUTINE
	TERMT	TERMINATE VIRTUAL TERMINAL INTERFACE
	TRMDRV	TERMINATE DEVICE DRIVER
	TRMUSR	TERMINATE USER
	UIS	USR INTERFACE SERVICES
	UIS/FLWIN	FILL WINDOW INFORMATION
	UIS/FLWNS	FILL WINDOW MANAGER STRUCTURE
	UIS/PRCIN	PROCESS INPUT
	UIS/PRCWN	PROCESS WINDOW
	UIS/STRTA	START APPLICATION
	UIS/STRTP	START PHYSICAL DEVICE
	ULKFPD	UNLINK FPD

FORM PROCESSOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
--------------------------	-------------------------	----------------------------

TIME

CMPFLD	COMPUTE FIELD	
CMPFLD/EV	EVALUATE FIELD EXPRESSION	
INSCR	INPUT SCREEN	
OISCR/ADD	ADD COMMAND TO BUFFER	
OISCR/CMP	COMPUTE ALL CALCULATED FIELDS	
OISCR/CNG	CHECK FOR AND PROCESS CHANGE MESSAGE REQUESTS	
OISCR/DSP	DISPLAY SCREEN	
OISCR/EVT	EMPTY VTI BUFFER	
OISCR/FVT	FILL VTI BUFFER	
OISCR/GAT	GET ATTRIBUTE INFO	
OISCR/PRO	PROCESS FIELD	
OISCR/PRO	PROCESS WINDOW	
OISCR/RST	RESET INPUT FLAGS	
OISCR/RST	RESET TEMPORAY ATTRIBUTES	
OISCR/SET	SET WINDOW	
ONWISC	OUTPUT (NO WAIT) / INPUT SCREEN	
OUTSCR	OUTPUT SCREEN	

UISFM

UIS	USR INTERFACE SERVICES	
UIS/FLWIN	FILL WINDOW INFORMATION	
UIS/FLWNS	FILL WINDOW MANAGER STRUCTURE	
UIS/PRCIN	PRCESS INPUT	
UIS/PRCWN	PRCESS WINDOW	
UIS/STRTA	START APPLICATION	
UIS/STRTP	START PHYSICAL DEVICE	

VTICOM

FORM PROCESSOR Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	INSCR	INPUT SCREEN
	OISCR/ADD	ADD COMMAND TO BUFFER
	OISCR/CMP	COMPUTE ALL CALCULATED FIELDS
	OISCR/CNG	CHECK FOR AND PROCESS CHANGE MESSAGE REQUESTS
	OISCR/DSP	DISPLAY SCREEN
	OISCR/EVT	EMPTY VTI BUFFER
	OISCR/FVT	FILL VTI BUFFER
	OISCR/GAT	GET ATTRIBUTE INFO
	OISCR/PRO	PROCESS FIELD
	OISCR/PRO	PROCESS WINDOW
	OISCR/RST	RESET INPUT FLAGS
	OISCR/RST	RESET TEMPORAY ATTRIBUTES
	OISCR/SET	SET WINDOW
	ONWISC	OUTPUT (NO WAIT) / INPUT SCREEN
	OUTSCR	OUTPUT SCREEN
	PDVOTP	PUT DEVICE OUTPUT

3.10.6 Where External Routine Used List

The following lists each external function or routine listed in 3.10.3 and all the documented modules which call it. The purpose of each module is listed as well.

FORM PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
ABORT		OPNFRM/BRPBUILD RELATIVE POSITION NODE
ABS		ADDELM ADD ELEMENT ADJSTR ADJUST FORM PROCESSOR STRUSCTURE COPFLD/CPYINTERNAL COPY FIELD CURPOS/FNDFIND CURSOR POSITION OISCR/PROCPROCESS FIELD OISCR/PROCPROCESS WINDOW OPNFRM/PARPROCESS ARRAY
ATOI		MAKAP MAKE APLICATION STRUTURE RMVAP REMOVE APPLICATION SYSMSG SYSTEM MESSAGE ROUTINE
BLDCMD		GDVINP GET DEVICE INPUT MAKUSR MAKE USER MONITR/MAIMAIN MODULE FOR MONITOR/UIS/FP PROCESS OISCR/EVTBEMPTY VTI BUFFER
BLEN		CANITM CANONICALIZE ITEM CMPFLD COMPUTE FIELD CMPFLD/EVAEVALUATE FIELD EXPRESSION COPFLD/CPYINTERNAL COPY FIELD GDATA/GETBGET BUFFER GDATLN/GBUGET BUFFER LENGTH OISCR/EVTBEMPTY VTI BUFFER OISCR/PROCPROCESS FIELD

FORM PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
		OISCR/RSTIRESET INPUT FLAGS OPNFRM/BFLBUILD FIELD DEFAULT BUFFER PDATA/PUTBPUT BUFFER
CALLOC	MAKAP	MAKE APPLICATION STRUTURE
	MAKFPD	MAKE FORM PROCESSOR DATA (LOGICAL DEVICE STRUCTURE)
	MAKPD	MAKE PHYSICAL DEVICE STRUCTURE
	MAKUSR	MAKE USER
CBIT		COPFLD/CPYINTERNAL COPY FIELD DELFLD DELETE FIELD RMVAP REMOVE APPLICATION
CBPTR	CANITM	CANONICALIZE ITEM
	CMPFLD	COMPUTE FIELD
	CMPFLD/EVAEVALUATE	FIELD EXPRESSION
	COPFLD/CPYINTERNAL	COPY FIELD
	GDATA/GETBGET	BUFFER
	OISCR/EVTBEMPTY	VTI BUFFER
	OISCR/PROCPROCESS	FIELD
	OISCR/RSTIRESET	INPUT FLAGS
	PDATA/PUTBPUT	BUFFER
CFREE	MAKAP	MAKE APPLICATION STRUTURE
	MAKUSR	MAKE USER
	RMVAP	REMOVE APPLICATION
	RMVFPD	REMOVE FORM PROCESSOR DATA STRUCTURE

FORM PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
	RMVPD TRMUSR	REMOVE PHYSICAL DEVICE DATA STRUCTURE TERMINATE USER
DBCLSE	MONITR/MAIMAIN MODULE FOR MONITOR/UIS/FP PROCESS	
DBCOM	UIS/STRTAPSTART APPLICATION	
DBCUPR	UIS	USR INTERFACE SERVICES
DBGAPD	UIS/STRTAPSTART APPLICATION	
DBOPEN	MONITR/MAIMAIN MODULE FOR MONITOR/UIS/FP PROCESS	
DOATTR	PRNFLD	PRINT FIELD
DOITEM	PRNFLD	PRINT FIELD

FORM PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
DOWIND	PRNFLD	PRINT FIELD
DSPMSG	INSCR	INPUT SCREEN
	OISCR/FVTBFILL	VTI BUFFER
	RMVAP	REMOVE APPLICATION
ERRPRO	SYMSG	SYSTEM MESSAGE ROUTINE
FCLOSE	GDVINP	GET DEVICE INPUT
	MONITR/MAIMAIN	MODULE FOR MONITOR/UIS/FP PROCESS
	OPNFRM	OPEN FORM
	PDVOTP	PUT DEVICE OUTPUT
	RMVPD	REMOVE PHYSICAL DEVICE DATA STRUCTURE
	UIS/STRTPDSTART	PHYSICAL DEVICE
FEOF	MONITR/MAIMAIN	MODULE FOR MONITOR/UIS/FP PROCESS
	OPNFRM/BDBBUILD	DEFAULT BUFFER
FERROR	MONITR/MAIMAIN	MODULE FOR MONITOR/UIS/FP PROCESS
	OPNFRM/BDBBUILD	DEFAULT BUFFER
	OPNFRM/BRPBUILD	RELATIVE POSITION NODE
	OPNFRM/BTBBUILD	TEXT BUFFER

FORM PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
---------------------------	-------------------------	----------------------------

FFBCA

COPFLD/CPYINTERNAL COPY FIELD
STUPFP SET UP FORM PROCESSOR DATA STRUCTURES
UIS/STRTPDSTART APPLICATION
UIS/STRTPDSTART PHYSICAL DEVICE

FOPEN

FNDMSG/OUNOPEN USER MESSAGE FILE
MONITR/MAIMAIN MODULE FOR MONITOR/UIS/FP PROCESS
OPNFRM OPEN FORM
UIS/STRTPDSTART PHYSICAL DEVICE

FPRINTF

MONITR/MAIMAIN MODULE FOR MONITOR/UIS/FP PROCESS

FREAD

FNDMSG/CODCODE SEARCH
MONITR/MAIMAIN MODULE FOR MONITOR/UIS/FP PROCESS
OPNFRM/BRPBUILD RELATIVE POSITION NODE
OPNFRM/PDRPROCESS FIELD RECORD
OPNFRM/PFRPROCESS FORM RECORD
OPNFRM/PTRPROCESS TEXT RECORD

FREE

CMPFLD COMPUTE FIELD
CMPFLD/EVAEVALUATE FIELD EXPRESSION
COPFLD/CPYINTERNAL COPY FIELD
DELFLD DELETE FIELD
DELFLD/DELDELETE EXPRESSION

FORM PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
	GETCUR	GET CURSOR POSITION
	OPNFRM/BDBBUILD	DEFAULT BUFFER
	PTHPTR	GET PATH POINTER
	RSVEXP/BLDBUILD	EXPRESSION TREE
	UIS/STRTPDSTART	PHYSICAL DEVICE
FREMSG	INSCR	INPUT SCREEN
	RMVFPD	REMOVE FORM PROCESSOR DATA STRUCTURE
FSEARCH	UIS/STRTPDSTART	PHYSICAL DEVICE
FSEEK	MONITR/MAIMAIN	MODULE FOR MONITOR/UIS/FP PROCESS
FTELL	GDVINP	GET DEVICE INPUT
	MONITR/MAIMAIN	MODULE FOR MONITOR/UIS/FP PROCESS
FWRITE	GDVINP	GET DEVICE INPUT
	MONITR/MAIMAIN	MODULE FOR MONITOR/UIS/FP PROCESS
	PDVOTP	PUT DEVICE OUTPUT
GETC	OPNFRM/BDBBUILD	DEFAULT BUFFER
	OPNFRM/BTBBUILD	TEXT BUFFER

FORM PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
GETW		MONITR/MAIMAIN MODULE FOR MONITOR/UIS/FP PROCESS
GVTICMD	GDVINP	GET DEVICE INPUT
	MAKUSR	MAKE USER
		MONITR/MAIMAIN MODULE FOR MONITOR/UIS/FP PROCESS
		OISCR/EVTBEMPTY VTI BUFFER
GVTINW	CALLFP	CALL FP ROUTINES
INITAL		MONITR/MAIMAIN MODULE FOR MONITOR/UIS/FP PROCESS
ISALNUM	PTHPTR	GET PATH POINTER
ISCNTRL		OPNFRM/BDBBUILD DEFAULT BUFFER
		OPNFRM/BTBUILD TEXT BUFFER
ISDIGIT	CMPFLD/EVAEVALUATE	FIELD EXPRESSION
	PTHPTR	GET PATH POINTER

FORM PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
	RSVEXP/BLDBUILD	EXPRESSION TREE
ISEND	UIS/STRTAPSTART	APPLICATION
	UIS/STRTPDSTART	PHYSICAL DEVICE
ISPRINT	OISCR/EVTBEMPTY	VTI BUFFER
	PDATA/PUTBPUT	BUFFER
	PMSGLS	PUT MESSAGE LINE STRING
LOCALTIME	OISCR/DSPSDISPLAY	SCREEN
MALLOC	CMPFLD/EVAEVALUATE	FIELD EXPRESSION
	COPFLD/CPYINTERNAL	COPY FIELD
	GETCUR	GET CURSOR POSITION
	MAKFLD	MAKE FIELD
	OPNFRM/BDBBUILD	DEFAULT BUFFER
	OPNFRM/BFLBUILD	FIELD DEFAULT BUFFER
	OPNFRM/BRPBUILD	RELATIVE POSITION NODE
	OPNFRM/BTBBUILD	TEXT BUFFER
	OPNFRM/PITPROCESS	ITEM
	OPNFRM/PTRPROCESS	TEXT RECORD
	PMSGLS	PUT MESSAGE LINE STRING
	PTHPTR	GET PATH POINTER
	RSVEXP/BLDBUILD	EXPRESSION TREE
MATOI		

FORM PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
---------------------------	-------------------------	----------------------------

	CALLFP	CALL FP ROUTINES
	OISCR/EVTBEMPTY	VTI BUFFER
	UIS/PRCINPPRCESS	INPUT
	UIS/PRCWNDPRCESS	WINDOW

MAX

	ADDELM	ADD ELEMENT
	COPFRM	COPY FORM
	MONITR/MAIMAIN	MODULE FOR MONITOR/UIS/FP PROCESS
	OPNFRM/PARPROCESS	ARRAY
	POSCUR	POSITION CURSOR
	RMVPAG	REMOVE PAGE
	UIS/PRCINPPRCESS	INPUT

MEMCMP

	CALLFP	CALL FP ROUTINES
	FNDMSG/CODCODE	SEARCH
	INSCR	INPUT SCREEN
	MONITR/MAIMAIN	MODULE FOR MONITOR/UIS/FP PROCESS
	OPNFRM/PITPROCESS	ITEM
	PDATA/PUTBPUT	BUFFER
	PDVOTP	PUT DEVICE OUTPUT
	TRMDRV	TERMINATE DEVICE DRIVER
	TRMUSR	TERMINATE USER
	UIS	USR INTERFACE SERVICES
	UIS/FLWNSTFILL	WINDOW MANAGER STRUCTURE
	UIS/PRCINPPRCESS	INPUT
	UIS/PRCWNDPRCESS	WINDOW
	UIS/STRTPSTART	APPLICATION
	UIS/STRTPDSTART	PHYSICAL DEVICE

MEMCPY

	ADDELM	ADD ELEMENT
	ADDFRM	ADD FORM TO WINDOW

FORM PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
	CALLFP	CALL FP ROUTINES
	CHGLDV	CHANGE LOGICAL DEVICE
	CLSFRM	CLOSE FORM
	CLSLDV	CLOSE LOGICAL DEVICE
	CMPFLD	COMPUTE FIELD
	CMPFLD/EVAEVALUATE	FIELD EXPRESSION
	COPFLD/CPYINTERNAL	COPY FIELD
	FNDMSG/CODCODE	SEARCH
	FNDMSG/OUOPEN	USER MESSAGE FILE
	GDATA	GET DATA
	GDATA/GETBGET	BUFFER
	GDATLN	GET DATA LENGTH
	GDVINP	GET DEVICE INPUT
	GETATT	GET ATTRIBUTE
	GETBAK	GET BACKGROUND ATTRIBUTE
	GETCUR	GET CURSOR POSITION
	GETCUR/CONCONCATENATE	STRING TO CURRENT NAME
	GPAGE	GET PAGE
	GWINDO	GET WINDOW
	INITVT	INITIAL VIRTUAL TERMINAL INTERFACE
	INQLDV	INQUIRE LOGICAL DEVICE
	MONITR/MAIMAIN	MODULE FOR MONITOR/UIS/FP PROCESS
	OISCR/ADDCA	COMMAND TO BUFFER
	OISCR/EVTBEMPTY	VTI BUFFER
	OISCR/RSTIRESET	INPUT FLAGS
	ONWISC	OUTPUT (NO WAIT) / INPUT SCREEN
	OPNFRM	OPEN FORM
	OPNFRM/BFLBUILD	FIELD DEFAULT BUFFER
	OPNFRM/PITPROCESS	ITEM
	OPNLDV	OPEN LOGICAL DEVICE
	OUTSCR	OUTPUT SCREEN
	PARFQN	PARSE FULLY QUALIFIED NAME
	PDATA	PUT FORM DATA
	PDATA/PUTBPUT	BUFFER
	PDVOTP	PUT DEVICE OUTPUT
	PUTATT	PUT ATTRIBUTES
	PUTBAK	PUT BACKGROUND ATTRIBUTES
	PUTCUR	PUT CURSOR
	PUTLOC	PUT LOCATION
	RMVPAG	REMOVE PAGE

FORM PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
	RPLFRM	REPLACE FORM
	TERMVT	TERMINATE VIRTUAL TERMINAL INTERFACE
	TRMDRV	TERMINATE DEVICE DRIVER
	TRMUSR	TERMINATE USER
	UIS	USR INTERFACE SERVICES
MEMSET		
	CALLFP	CALL FP ROUTINES
	CANITM	CANONICALIZE ITEM
	CMPFLD	COMPUTE FIELD
	FNDMSG	FIND MESSAGE
	GDVINP	GET DEVICE INPUT
	GETATT	GET ATTRIBUTE
	GETBAK	GET BACKGROUND ATTRIBUTE
	GETCUR	GET CURSOR POSITION
	GPAGE	GET PAGE
	MONITR/MAIMAIN	MODULE FOR MONITOR/UIS/FP PROCESS
	OISCR/EVTBEMPTY	VTI BUFFER
	PARFQN	PARSE FULLY QUALIFIED NAME
	PDVOTP	PUT DEVICE OUTPUT
	PMSGLS	PUT MESSAGE LINE STRING
	TRMDRV	TERMINATE DEVICE DRIVER
	TRMUSR	TERMINATE USER
	UIS	USR INTERFACE SERVICES
	UIS/FLWNSTFILL	WINDOW MANAGER STRUCTURE
	UIS/PRCWNDPROCESS	WINDOW
MIN		
	CALLFP	CALL FP ROUTINES
	MONITR/MAIMAIN	MODULE FOR MONITOR/UIS/FP PROCESS
	OISCR/PROCPROCESS	FIELD
	POSCUR	POSITION CURSOR
	PTHPTR/FIEMATCH	FIELD
	UIS/PRCINPPRCESS	INPUT

FORM PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
NSEND	CALLFP	CALL FP ROUTINES
	MONITR/MAIMAIN	MODULE FOR MONITOR/UIS/FP PROCESS
	PDVOTP	PUT DEVICE OUTPUT
	TRMDRV	TERMINATE DEVICE DRIVER
OBIND	DBCFCNC	CHECK FUNCTION
	DBCROL	CHECK ROLE
ODFINN	DBCFCNC	CHECK FUNCTION
	DBCROL	CHECK ROLE
OEXEC	DBCFCNC	CHECK FUNCTION
	DBCROL	CHECK ROLE
OFETCH	DBCFCNC	CHECK FUNCTION
	DBCROL	CHECK ROLE
OSQL3	DBCFCNC	CHECK FUNCTION
	DBCROL	CHECK ROLE

FORM PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
PBPTR		COPFLD/CPYINTERNAL COPY FIELD OISCR/RSTIRESET INPUT FLAGS
PFINP	INSCR	INPUT SCREEN
PRINTF	PRNAP PRNFLD PRNPD PRNUID PRNUSR	PRINT APPLICATION PRINT FIELD PRINT PHYSICAL DEVICE PRINT UID PRINT USER
PUTC		MONITR/MAIMAIN MODULE FOR MONITOR/UIS/FP PROCESS
PUTW	GDVINP	GET DEVICE INPUT MONITR/MAIMAIN MODULE FOR MONITOR/UIS/FP PROCESS
RCV		MONITR/MAIMAIN MODULE FOR MONITOR/UIS/FP PROCESS
REWIND		

FORM PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
	FNDMSG/CODCODE	SEARCH
SBIT	COPFLD/CPYINTERNAL	COPY FIELD
	MAKAP	MAKE APPLICATION STRUCTURE
	STUPFP	SET UP FORM PROCESSOR DATA STRUCTURES
SIGABT	TRMUSR	TERMINATE USER
	UIS/STRTAPSTART	APPLICATION
	UIS/STRTPDSTART	PHYSICAL DEVICE
SNDVTI	CALLFP	CALL FP ROUTINES
SPRINTF	CALLFP	CALL FP ROUTINES
	CMPFLD/EVAEVALUATE	FIELD EXPRESSION
	FNDMSG	FIND MESSAGE
	FNDMSG/OUNOPEN	USER MESSAGE FILE
	GETCUR	GET CURSOR POSITION
	MAKUSR	MAKE USER
	MONITR/MAIMAIN	MODULE FOR MONITOR/UIS/FP PROCESS
	OISCR/DSPSDISPLAY	SCREEN
	OISCR/FVTBFILL	VTI BUFFER
	OISCR/PROCPROCESS	FIELD
	OISCR/PROCPROCESS	WINDOW
	OPNFRM	OPEN FORM
	RMVAP	REMOVE APPLICATION
	RMVFPD	REMOVE FORM PROCESSOR DATA STRUCTURE
	UIS	USR INTERFACE SERVICES
	UIS/FLWINFILL	WINDOW INFORMATION

FORM PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
---------------------------	-------------------------	----------------------------

UIS/FLWNSTFILL WINDOW MANAGER STRUCTURE
UIS/PRCINPPRCESS INPUT
UIS/PRCWNDPRCESS WINDOW
UIS/STRTAPSTART APPLICATION
UIS/STRTPDSTART PHYSICAL DEVICE

STRASN

COPFLD/CPYINTERNAL COPY FIELD
MABSAT MAP ABSOLUTE ATTRIBUTE
OISCR/EVTBEMPTY VTI BUFFER
OISCR/GATIGET ATTRIBUTE INFO
OPNFRM/BRPBUILD RELATIVE POSITION NODE
OPNFRM/PWIPROCESS WINDOW
PTHPTR GET PATH POINTER
RMVPAG REMOVE PAGE
RSVATT RESOLVE ATTRIBUTE
RSVATT/RSVRESOLVE REST
STUPFP SET UP FORM PROCESSOR DATA STRUCTURES

STRCAT

OISCR/PROCPROCESS FIELD

STRCHR

MONITR/MAIMAIN MODULE FOR MONITOR/UIS/FP PROCESS
PARFQN PARSE FULLY QUALIFIED NAME
RSVEXP/BLDBUILD EXPRESSION TREE

STRCMP

ADDELM ADD ELEMENT
FNDFLD FIND FIELD
GATDEF GET ATTRIBUTE DEFINITION
GOFPTR GET OPEN FROM POINTER

FORM PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
	MONITR/GETGET	PHYSICAL DEVICE
	MONITR/MAIMAIN	MODULE FOR MONITOR/UIS/FP PROCESS
	OPNFRM/PFRPROCESS	FORM RECORD
	PTHPTR/FIEMATCH	FIELD
	PTHPTR/FOUHAS	ANYTHING BEEN FOUND?
	SFPDAP	SET FORM PROCESSOR DATA STRUCTURE FOR APPLICATION
	UIS/PRCINPPRCESS	INPUT
	UIS/STRTPDSTART	PHYSICAL DEVICE
STRCPY	MAKFLD	MAKE FIELD
STRLEN	CALLFP	CALL FP ROUTINES
	CMPFLD	COMPUTE FIELD
	CMPFLD/EVAEVALUATE	FIELD EXPRESSION
	FNDMSG	FIND MESSAGE
	GDVINP	GET DEVICE INPUT
	GETATT	GET ATTRIBUTE
	GETBAK	GET BACKGROUND ATTRIBUTE
	GETCUR/CONCONCATENATE	STRING TO CURRENT NAME
	GPAGE	GET PAGE
	MAKUSR	MAKE USER
	OISCR/FVTBFILL	VTI BUFFER
	OISCR/PROCPROCESS	FIELD
	OISCR/PROCPROCESS	WINDOW
	OPNFRM/PITPROCESS	ITEM
	PARFQN	PARSE FULLY QUALIFIED NAME
	PDVOTP	PUT DEVICE OUTPUT
	PMSGLS	PUT MESSAGE LINE STRING
	RMVFPD	REMOVE FORM PROCESSOR DATA STRUCTURE
	TRMDRV	TERMINATE DEVICE DRIVER
	TRMUSR	TERMINATE USER
	UIS	USR INTERFACE SERVICES
	UIS/PRCINPPRCESS	INPUT

FORM PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
	UIS/STRTPDSTART	PHYSICAL DEVICE
	UIS/STRTAPSTART	APPLICATION
STRNCMP	FNDMSG SYSMSG	FIND MESSAGE SYSTEM MESSAGE ROUTINE
STRNCPY	ESCPY	EXTERNAL STRING COPY
STRNLOC	CANITM	CANONICALIZE ITEM
STRNUPC	CANITM	CANONICALIZE ITEM
STRRCHR	PARFQN	PARSE FULLY QUALIFIED NAME
STRUPC	CLSFRM COPFRM GATDEF OPNFRM	CLOSE FORM COPY FORM GET ATTRIBUTE DEFINITION OPEN FORM
TIME		

FORM PROCESSOR Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
---------------------------	-------------------------	----------------------------

OISCR/DSPSDISPLAY SCREEN

TOUPPER

PTHPTR GET PATH POINTER

TRMNAT

MONITR/MAIMAIN MODULE FOR MONITOR/UIS/FP PROCESS

3.10.7 Main Program Parts List

The following lists each Main Program listed in 3.10.1 and all the modules which are called either by that module itself or by any of the documented modules which it calls. It is possible for a non-main module to be listed more than once if it is called by multiple modules. The called modules, in this case known as program parts, are marked as to whether they are documented here. If so, the phrase "well-defined module" appears by the module name, if not it is an "external routine". The Purpose of the Main Program module is listed as well.

FORM PROCESSOR Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
MONITR/MAIN	Purpose-->	MAIN MODULE FOR MONITOR/UIS/FP PROCESS
	ABORT	External routine
	ABS	External routine
	ACRPOS	Well-defined module
	ADDELM	Well-defined module
	ADDFRM	Well-defined module
	ATOI	External routine
	BLDCMD	External routine
	BLEN	External routine
	CALLFP	Well-defined module
	CALLOC	External routine
	CANITM	Well-defined module
	CBIT	External routine
	CBPTR	External routine
	CFREE	External routine
	CHGLDV	Well-defined module
	CHGPRC	Well-defined module
	CLSFRM	Well-defined module
	CLSLDV	Well-defined module
	CMPFLD	Well-defined module
	CMPFLD/EVAL	Well-defined module
	COPFLD	Well-defined module
	COPFLD/CPYFLD	Well-defined module
	COPFRM	Well-defined module
	CURPOS	Well-defined module
	CURPOS/FNDCEP	Well-defined module
	DBCFCNC	Well-defined module
	DBCLSE	External routine
	DBCOM	External routine
	DBCROL	Well-defined module
	DBCUPR	External routine
	DBGAPD	External routine
	DBOPEN	External routine
	DELFLD	Well-defined module
	DELFLD/DELEXP	Well-defined module
	DSPMSG	External routine
	ERRPRO	External routine
	ESCPY	Well-defined module
	FCLOSE	External routine

FORM PROCESSOR Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
	FEOF	External routine
	FERROR	External routine
	FFBCA	External routine
	FNDFLD	Well-defined module
	FNDMSG	Well-defined module
	FNDMSG/CODSCH	Well-defined module
	FNDMSG/OUMSGF	Well-defined module
	FNFPWN	Well-defined module
	FOPEN	External routine
	FPRINTF	External routine
	FREAD	External routine
	FREE	External routine
	FREMSG	External routine
	FSEARCH	External routine
	FSEEK	External routine
	FTELL	External routine
	FUISWN	Well-defined module
	FWRITE	External routine
	GATDEF	Well-defined module
	GDATA	Well-defined module
	GDATA/GETBUF	Well-defined module
	GDATLN	Well-defined module
	GDATLN/GBUFLN	Well-defined module
	GDVINP	Well-defined module
	GETATT	Well-defined module
	GETBAK	Well-defined module
	GETC	External routine
	GETCUR	Well-defined module
	GETCUR/CONCAT	Well-defined module
	GETW	External routine
	GOFPTR	Well-defined module
	GPAGE	Well-defined module
	GVTICMD	External routine
	GVTINW	External routine
	GWINDO	Well-defined module
	INITAL	External routine
	INITVT	Well-defined module
	INQLDV	Well-defined module
	ISALNUM	External routine
	ISCNTRL	External routine

FORM PROCESSOR Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
	ISDIGIT	External routine
	ISEND	External routine
	ISPRINT	External routine
	LOCALTIME	External routine
	MABSAT	Well-defined module
	MAKAP	Well-defined module
	MAKFLD	Well-defined module
	MAKFPD	Well-defined module
	MAKPD	Well-defined module
	MAKUSR	Well-defined module
	MALLOC	External routine
	MATOI	External routine
	MAX	External routine
	MEMCMP	External routine
	MEMCPY	External routine
	MEMSET	External routine
	MIN	External routine
	MONITR/GETPD	Well-defined module
	NSEND	External routine
	OBIND	External routine
	ODFINN	External routine
	OEXEC	External routine
	OFETCH	External routine
	OISCR/ADDCMD	Well-defined module
	OISCR/CMPALL	Well-defined module
	OISCR/DSPSCR	Well-defined module
	OISCR/FVTBUF	Well-defined module
	OISCR/GATINF	Well-defined module
	OISCR/PROCFLD	Well-defined module
	OISCR/PROCWIN	Well-defined module
	OISCR/RSTINP	Well-defined module
	OISCR/SETWIN	Well-defined module
	ONWISC	Well-defined module
	OPNFRM	Well-defined module
	OPNFRM/BDBUFF	Well-defined module
	OPNFRM/BFLDDB	Well-defined module
	OPNFRM/BRPNOD	Well-defined module
	OPNFRM/BTBUFF	Well-defined module
	OPNFRM/PARY	Well-defined module
	OPNFRM/PDREC	Well-defined module

FORM PROCESSOR Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
	OPNFRM/PFREC	Well-defined module
	OPNFRM/PFRM	Well-defined module
	OPNFRM/PITM	Well-defined module
	OPNFRM/PTREC	Well-defined module
	OPNFRM/PWIN	Well-defined module
	OPNLDV	Well-defined module
	OSQL3	External routine
	OUTSCR	Well-defined module
	PARFQN	Well-defined module
	PBPTR	External routine
	PDATA	Well-defined module
	PDATA/PUTBUF	Well-defined module
	PDVOTP	Well-defined module
	PMSGLC	Well-defined module
	PMSGLS	Well-defined module
	POSCUR	Well-defined module
	POSCUR/FNFITM	Well-defined module
	PTHPTR	Well-defined module
	PTHPTR/ARRAY	Well-defined module
	PTHPTR/FIELD	Well-defined module
	PTHPTR/FORM	Well-defined module
	PTHPTR/FOUND	Well-defined module
	PTHPTR/ITEM	Well-defined module
	PTHPTR/WINDOW	Well-defined module
	PUTATT	Well-defined module
	PUTATT/AABSAT	Well-defined module
	PUTBAK	Well-defined module
	PUTC	External routine
	PUTCUR	Well-defined module
	PUTLOC	Well-defined module
	PUTW	External routine
	RCV	External routine
	REWIND	External routine
	RMVAP	Well-defined module
	RMVFPD	Well-defined module
	RMVPAG	Well-defined module
	RMVPD	Well-defined module
	RPLFRM	Well-defined module
	RSVATT	Well-defined module
	RSVATT/RSVRST	Well-defined module

FORM PROCESSOR Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
	RSVEXP	Well-defined module
	RSVEXP/BLDEXP	Well-defined module
	SBIT	External routine
	SFPDAP	Well-defined module
	SIGABT	External routine
	SNDVTI	External routine
	SPRINTF	External routine
	STRASN	External routine
	STRCAT	External routine
	STRCHR	External routine
	STRCMP	External routine
	STRCPY	External routine
	STRLEN	External routine
	STRNCMP	External routine
	STRNCPY	External routine
	STRNLOC	External routine
	STRNUPC	External routine
	STRRCHR	External routine
	STRUPC	External routine
	STUPFP	Well-defined module
	SYSMSG	Well-defined module
	TERMT	Well-defined module
	TIME	External routine
	TOUPPER	External routine
	TRMDRV	Well-defined module
	TRMNAT	External routine
	TRMUSR	Well-defined module
	UIS	Well-defined module
	UIS/FLWINF	Well-defined module
	UIS/FLWNST	Well-defined module
	UIS/PRCINP	Well-defined module
	UIS/PRCWND	Well-defined module
	UIS/STRTAP	Well-defined module
	UIS/STRTPD	Well-defined module
	ULKFPD	Well-defined module

FORM PROCESSOR Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
PRNDSP		Purpose-->PRINT DISPLAY LIST
	DOATTR	External routine
	DOITEM	External routine
	DOWIND	External routine
	PRINTF	External routine
	PRNFLD	Well-defined module

FORM PROCESSOR Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
PRNOPN		Purpose-->PRINT OPEN LIST
	DOATTR	External routine
	DOITEM	External routine
	DOWIND	External routine
	PRINTF	External routine
	PRNFLD	Well-defined module

FORM PROCESSOR Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
PRNUID		Purpose-->PRINT UID
	PRINTF	External routine
	PRNAP	Well-defined module
	PRNPD	Well-defined module

FORM PROCESSOR Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
PRNUSR		Purpose-->PRINT USER
	PRINTF	External routine
	PRNAP	Well-defined module
	PRNPD	Well-defined module

3.10.8 Module Documentation

The following documentation describes information which is specific to each individual module being documented in this specification as listed in section 3.10.2. It provides a compact way of getting information that would be otherwise buried within each module's source code.

The specific items in this module documentation have the following meanings:

NAME:	Name of program Module.
PURPOSE:	Purpose of Module as detailed in the source code.
LANGUAGE:	Programming language source code is written in. The choices are: VAX-11 FORTRAN C (I/S-1 Workbench 'C') VAX-11 COBOL
MODULE TYPE:	Whether a Program, Subroutine, or Function.
SOURCE FILE:	Name of Source File from file specification.
SOURCE FILE TYPE:	Source File Extension from file specification.
HOST:	Whether this is a host-dependent routine (VAX or IBM) or blank if host-independent.
SUBSYSTEM:	IISS sub-system this file resides in.
SUBDIRECTORY:	Sub-directory of that subsystem in which this file resides.
DOCUMENTATION GROUP:	Name of documentation group of which this source file is a member.
DESCRIPTION:	A description of the module as obtained from the source code.
ARGUMENTS:	The arguments with which this routine is called if it is a Subroutine or a Function.
INCLUDE FILES:	A list of all the files that are included into this module as well as their purposes.

ROUTINES CALLED: Subroutines or Functions, either documented or external, called by this module, if any.

CALLED DIRECTLY BY: The documented routines which call this module, if any.

USED IN MAIN PROGRAM(S): The documented Main Programs which contain this module in their parts list according to the list in section 3.10.7.

The Module Documentation is arranged alphabetically according to Module Name.

FORM PROCESSOR Module Documentation

NAME: ACRPOS
PURPOSE: ABSOLUTIZE CURSOR POSITION OF FIELD
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: ACRPOS
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
VOID ACRPOS(DP,ABSPOS)
    FIELD    *DP;
    POSITION  *ABSPOS;
```

INPUTS/OUTPUTS:

INPUTS:

DP - FIELD WHOSE ROW AND COL WANT TO ABSOLUTIZED
ADDRESS OF STURCTURE FOR RETURNING VALJES OF:

ROW	ABSOLUTE
COL	ABSOLUTE

OUTPUTS:

STRUCTURE CONTAINING:

ABSOLUTE ROW OF FIELD
ABSOLUTE COL OF FIELD

DESCRIPTION

THIS MODULE ABSOLUTIZES A FIELD'S ROW AND COL BY GOING
BACK UP
CHILD PARENT TREE AND ADDING EACH SUCCESSIVE PARENT'S ROW
AND COL
TO SUM OF CHILDS'.

ARGUMENTS:

DP = FIELD *
ABSPOS = POSITION *

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA

CALLED DIRECTLY BY:

CURPOS/FND CP - FIND CURSOR POSITION
GETCUR - GET CURSOR POSITION

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: ADDELM
PURPOSE: ADD ELEMENT
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: ADDELM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
ADDELM(EPATH, PCOUNT, RCODE)
    EPATH  EPATH;
    INT *PCOUNT;
    CHAR RCODE[RCODE_LEN];
```

INPUTS:

EPATH - ARRAY TO ADD ELEMENT TO

OUTPUTS:

PCOUNT - INDEX OF ELEMENT ADDED
RCODE - RETURN CODE OF OPERATION

DESCRIPTION

ADDS AN ELEMENT TO AN OPEN ENDED ARRAY.

ARGUMENTS:

```
-----
EPATH =      EPATH
PCOUNT =     INT *
RCODE =      CHAR [RCODE_LEN ]
```

INCLUDE FILES:

```
-----
STDYTP - STANDARD TYPE DEFINITIONS
FPD    - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
```

ROUTINES CALLED:

```
-----
ESCPY - EXTERNAL STRING COPY
PTHPTR - GET PATH POINTER
MEMCPY
ABS
```

SYSMSG	- SYSTEM MESSAGE ROUTINE
MAX	
RSVEXP	- RESOLVE EXPRESSIONS
COPFLD	- COPY FIELD
STRCMP	
GOFPTR	- GET OPEN FROM POINTER

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: ADDFRM
PURPOSE: ADD FORM TO WINDOW
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FORTRAN VOID ()
SOURCE FILE: ADDFRM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
FORTRAN VOID ADDFRM(EWPATH, EFNAME, PAGEP, RCODE)
    EPATH EWPATH;
    ENAME EFNAME;
    INT *PAGEP;
    CHAR RCODE[];
```

INPUTS:

EWPATH - PATH NAME OF WINDOW TO ADD FORM TO
EFNAME - NAME OF FORM TO ADD TO WINDOW

OUTPUTS:

PAGEP - PAGE NUMBER OF ADDED FORM
RCODE - RETURN CODE

DESCRIPTION

ADDRFRM ADDS A FORM TO A WINDOW.

ARGUMENTS:

EWPATH =	EPATH
EFNAME =	ENAME
PAGEP =	INT *
RCODE =	CHAR []

INCLUDE FILES:

STDYTP	- STANDARD TYPE DEFINITIONS
FPD	- FORM PROCESSOR DATA
FPCODE	- FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

ESCPY	- EXTERNAL STRING COPY
PTHPTR	- GET PATH POINTER

SYSMSG - SYSTEM MESSAGE ROUTINE
COPFRM - COPY FORM
MEMCPY

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES
UIS - USR INTERFACE SERVICES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: ADJSTR
PURPOSE: ADJUST FORM PROCESSOR STRUSCTURE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: ADJSTR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

VOID GARPOS(DP)
FIELD *DP;

INPUTS/OUTPUTS:

INPUTS:

DP - FIELD WHOSE CHANGE IN ROW, COL, WIDTH, OR DEPTH
ADJUSTING FOR

OUTPUTS:

NONE

DESCRIPTION

THIS MODULE ADJUSTS PARENT ARRAY STRUCTURES FOR CHANGE IN
A WINDOW
ELIMENT OF ARRAY WHOSE LOCATION OR SIZE HAS CHANGED

ARGUMENTS:

DP = FIELD *

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA

ROUTINES CALLED:

ABS
ADJSTR - ADJUST FORM PROCESSOR STRUSCTURE

PS 620344200
30 September 1990

CALLED DIRECTLY BY:

ADJSTR - ADJUST FORM PROCESSOR STRUSCTURE

FORM PROCESSOR Module Documentation

NAME: CALLFP
PURPOSE: CALL FP ROUTINES
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: CALLFP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
VOID CALLFP(APPLCT, LOGCH, INPTREC, LENGTH);  
    NAME      APPLCT;  
    CHAN      LOGCH;  
    INPTREC   *INPTREC;  
    REGISTER INT LENGTH;
```

INPUTS/OUTPUTS:

INPUTS:

APPLCT - APPLICATION WHICH IS CALLING FORM PROCESSOR
LOGCH - CHANNEL ON WHICH THE APPLICATION IS CALLING
FP
INPTREC - INPUT PARAMETERS FROM THIS APPLICATION
LENGTH - LENGTH OF BUFFER CONTAINING INPUT PARAMETERS

OUTPUTS:

NONE

DESCRIPTION

PERFORMS REQUESTED FP CALLS USING INPUT PARAMETERS
FOUND IN
INPTREC.

ARGUMENTS:

APPLCT = NAME
LOGCH = CHAN
INPTREC = INPTREC *
LENGTH = INT

INCLUDE FILES:

STDTPY - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
NTM - NTM INTERFACE INCLUDE FILE

ROUTINES CALLED:

MEMSET
MATOI
RPLFRM - REPLACE FORM
RMVPAG - REMOVE PAGE
PUTLOC - PUT LOCATION
PUTBAK - PUT BACKGROUND ATTRIBUTES
PUTATT - PUT ATTRIBUTES
PARFQN - PARSE FULLY QUALIFIED NAME
GPAGE - GET PAGE
GVTINW
GETBAK - GET BACKGROUND ATTRIBUTE
GETATT - GET ATTRIBUTE
GDATA - GET DATA
CLSLDV - CLOSE LOGICAL DEVICE
CHGLDV - CHANGE LOGICAL DEVICE
ADDELM - ADD ELEMENT
SPRINTF
STRLEN
NSEND
MEMCMP
SYSMSG - SYSTEM MESSAGE ROUTINE
TERMT - TERMINATE VIRTUAL TERMINAL INTERFACE
SNDVTI
PUTCUR - PUT CURSOR
PMSGC - PUT MESSAGE LINE CODE
PMSG - PUT MESSAGE LINE STRING
PDATA - PUT FORM DATA
OUTSCR - OUTPUT SCREEN
OPNLDV - OPEN LOGICAL DEVICE
OPNFRM - OPEN FORM
ONWISC - OUTPUT (NO WAIT) / INPUT SCREEN
INQLDV - INQUIRE LOGICAL DEVICE
INITVT - INITIAL VIRTUAL TERMINAL INTERFACE
GWINDO - GET WINDOW
MIN
GETCUR - GET CURSOR POSITION
MEMCPY
GDATLN - GET DATA LENGTH
CLSFRM - CLOSE FORM
ADDFRM - ADD FORM TO WINDOW

CALLED DIRECTLY BY:

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

PS 620344200
30 September 1990

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: CANITM
PURPOSE: CANONICALIZE ITEM
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: CANITM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

VOID CANITM(DP)
FIELD *DP;

INPUTS:

DP - POINTER TO ITEM TO CANONICALIZE

DESCRIPTION

CANITM PERFORMS FORMAT CHANGES ON AN ITEM. POSSIBLE FORMAT
CHANGES ARE LEFT JUSTIFY, RIGHT JUSTIFY, UPPER CASE AND
LOWER CASE.

ARGUMENTS:

DP = FIELD *

INCLUDE FILES:

STDTPY - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA

ROUTINES CALLED:

CBPTR
BLEN
STRNLOC
MEMSET
STRNUPC

CALLED DIRECTLY BY:

CMPFLD - COMPUTE FIELD

PS 620344200
30 September 1990

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: CHGLDV
PURPOSE: CHANGE LOGICAL DEVICE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FORTRAN VOID ()
SOURCE FILE: CHGLDV
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
FORTRAN VOID CHGLDV(LDWNID,RCODE)
      INT          *LDWNID;
      CHAR          RCODE[];
```

INPUTS/OUTPUTS:

INPUTS:

LDWNID - LOGICAL DEVICES TOP WINDOW ID
ADDRESS OF:
RETURN CODE

OUTPUTS:

RCODE - RETURN CODE

DESCRIPTION

THIS MODULE CHANGES LOGICAL DEVICE TO WHICH APPLICATION
RUNS ON.

IF LOGICAL DEVICE NOT OPNED WILL RETURN NFPDSTRC ERROR.

ARGUMENTS:

LDWNID = INT *
RCODE = CHAR []

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

MEMCPY

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: CHGPRC
PURPOSE: CHANGE PRECEDENCE OF WINDOW OR LOGICAL
DEVICE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: CHGPRC
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
VOID CHGPRC(WDP,WFPD)
    FIELD *WDP;
    FPD    *WFPD;
```

INPUTS/OUTPUTS:

INPUTS:

WDP - POINTER TO WINDOW TO BE PUT AT TOPMOST PRECEDENCE
WFPD - POINTER TO LOG DEVICE OF WINDOW TO BE PUT AT
TOPMOST PRECEDENCE

OUTPUTS:

NONE

DESCRIPTION

THIS MODULE PUTS DEIGNATED WINDOW(OR DEVICE) AT BEGINNING
OF
LINK LIST OF WNDOWS AT ITS LEVEL (FSTFPD OR CONPTR)

ARGUMENTS:

WDP = FIELD *
WFPD = FPD *

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA

CALLED DIRECTLY BY:

UIS/PRCWND - PRCESS WINDOW

PS 620344200
30 September 1990

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: CLSFRM
PURPOSE: CLOSE FORM
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FORTRAN VOID ()
SOURCE FILE: CLSFRM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
FORTRAN VOID CLSFRM(EFNAME, CODEP)
    ENAME EFNAME;
    CHAR CODEP[];
```

INPUTS:

EFNAME - NAME OF FORM TO CLOSE

OUTPUTS:

CODEP - RETURN CODE

DESCRIPTION

USE CLSFRM TO CLOSE A FORM. MEMORY SPACE FOR THE FORM IS ELIMINATED SO THAT IT MAY BE ALLOCATED TO ANOTHER FORM.

NOTE: CLSFRM DOES NOT CLOSE A FORM IF THAT FORM IS CURRENTLY USED ANYWHERE ON THE DISPLAY LIST OR AS A SUBFORM TO A FORM ON THE OPEN LIST.

ARGUMENTS:

EFNAME = ENAME
CODEP = CHAR []

INCLUDE FILES:

STDTPY - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

ESCPY - EXTERNAL STRING COPY
STRUPC
GOFPTR - GET OPEN FROM POINTER

SYSMSG - SYSTEM MESSAGE ROUTINE
DEFLD - DELETE FIELD
MEMCPY

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: CLSLDV
PURPOSE: CLOSE LOGICAL DEVICE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FORTRAN VOID ()
SOURCE FILE: CLSLDV
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
FORTRAN VOID CLSLDV(LDWNID,RCODE)
      INT          *LDWNID;
      CHAR          RCODE[];
```

INPUTS/OUTPUTS:

INPUTS:

LDWNID - LOGICAL DEVICES TOP WINDOW ID
ADDRESS OF: RETURN CODE

OUTPUTS:

RCODE - RETURN CODE

DESCRIPTION

THIS MODULE CLOSE LOGICAL DEVICE. IF LOGICAL DEVICE NOT
OPENED WILL
RETURN NFPDSTRC OR CURFPDST ERRORS.

ARGUMENTS:

LDWNID = INT *
RCODE = CHAR []

INCLUDE FILES:

STDTPY - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

FNFPWN - FIND FORM PROCESSOR WINDOW
RMVFPD - REMOVE FORM PROCESSOR DATA STRUCTURE
MEMCPY

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: CMPFLD
PURPOSE: COMPUTE FIELD
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: CMPFLD
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

STATIC CHAR *CMPFLD(DP)
FIELD *DP;

INPUTS:

DP - POINTER TO FIELD TO BE COMPUTED

OUTPUTS:

RETURNS NULL OR ERROR CODE

DESCRIPTION

COMPUTES THE CURRENT VALUE OF A CALCULATED FIELD.

ARGUMENTS:

DP = FIELD *

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
TIME - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

CMPFLD/EVAL - EVALUATE FIELD EXPRESSION
STRLEN
BLEN
CANITM - CANONICALIZE ITEM
MEMCPY
SYSMSG - SYSTEM MESSAGE ROUTINE
FRFE
MEMSET
CBPTR

CALLED DIRECTLY BY:

OISCR/CMPALL - COMPUTE ALL CALCULATED FIELDS
RSVEXP - RESOLVE EXPRESSIONS

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: CMPFLD/EVAL
PURPOSE: EVALUATE FIELD EXPRESSION
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: CMPFLD
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

----- SYNOPSIS

```
    STATIC CHAR *EVAL(EP, RP)
        ENODE *EP;
        EXPVAL *RP;
```

INPUTS:

EP - POINTER TO EXPRESSION TO EVALUATE

OUTPUTS:

RP - POINTER TO RETURNED RESULT
RETURNS NULL OR ERROR CODE

DESCRIPTION

EVALUATES THE SPECIFIED FIELD EXPRESSION.

ARGUMENTS:

EP = ENODE *
RP = EXPVAL *

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
TIME - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

FREE
ISDIGIT
CMPFLD/EVAL - EVALUATE FIELD EXPRESSION
MEMCPY
SPRINTF
STRLEN
CBPTR

ESCPY - EXTERNAL STRING COPY
MALLOC
BLEN
SYSMSG - SYSTEM MESSAGE ROUTINE

CALLED DIRECTLY BY:

CMPFLD/EVA - EVALUATE FIELD EXPRESSION
CMPFLD - COMPUTE FIELD

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: COPFLD
PURPOSE: COPY FIELD
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: COPFLD
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
CHAR *COPFLD(DP, PAR_DP, ROW, COL, LNK_RDP, LNK_LDP)
    FIELD *PAR_DP;
    FIELD *DP;
    INT ROW, COL;
    FIELD **LNK_RDP, **LNK_LDP;
```

INPUTS:

NEW_DP - POINTER TO FIELD TO BE COPIED (IN OPEN LIST)
PAR_DP - POINTER TO PARENT OF NEWLY CREATED COPY
ROW - ROW WITHIN PARENT
COL - COLUMN WITHIN PARENT
LNK_RDP - POINTER TO BE SET TO POINT TO NEWLY CREATED COPY
RIGHT POINTER
LNK_LDP - POINTER TO BE SET TO POINT TO NEWLY CREATED COPY
LEFT POINTER

OUTPUTS:

ERROR CODE IS RETURNED IF ONE OCCURED ELSE A NULL IS RETURNED

DESCRIPTION

COPFLD COPIES A FORM FIELD AND ALL SUB FIELDS. THE COPY IS ALWAYS FROM THE OPEN LIST. A CHAIN OF FORWARD POINTERS IS KEPT TO LINK THE ACTIVE COPIES OF THE FORM, AS WELL AS A CHAIN OF BACKWARD POINTERS. SINCE DEFAULT FORM VALUES ARE MAINTAINED IN THE OPEN LIST, THE COPIED FORM HAS DEFAULT VALUES IN THE FIELDS.

ARGUMENTS:

DP = FIELD *

PAR_DP = FIELD *
LNK_RDP = FIELD **
LNK_LDP = FIELD **

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
BITS - INCLUDE FILE FOR BIT MANIPULATION ROUTINES

ROUTINES CALLED:

COPFLD/CPYFLD - INTERNAL COPY FIELD
RSVATT - RESOLVE ATTRIBUTE

CALLED DIRECTLY BY:

ADDELM - ADD ELEMENT
COPFRM - COPY FORM
OPNFRM/PFR - PROCESS FORM

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: COPFLD/CPYFLD
PURPOSE: INTERNAL COPY FIELD
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: COPFLD
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

DESCRIPTION

RECURSIVE ROUTINE CALLED TO COPY A FIELD AND ALL ITS
SUBFIELDS

ARGUMENTS:

DP = FIELD *
PAR_DP = FIELD *
LNK_RDP = FIELD **
LNK_LDP = FIELD **

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
BITS - INCLUDE FILE FOR BIT MANIPULATION ROUTINES

ROUTINES CALLED:

CBIT
ABS
COPFLD/CPYFLD - INTERNAL COPY FIELD
SBIT
FFBCA
PBPTR
CBPTR
MEMCPY
FREE
BLEN
STRASN
SYSMSG - SYSTEM MESSAGE ROUTINE
MALLOC

CALLED DIRECTLY BY:

COPFLD/CPYFLD - INTERNAL COPY FIELD
COPFLD - COPY FIELD

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: COPFRM
PURPOSE: COPY FORM
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: COPFRM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
CHAR *COPFRM(EFNAME, PARPTR, NXTPTR, PRVPTR)
    ENAME EFNAME;
    FIELD *PARPTR, **NXTPTR, **PRVPTR;
```

DESCRIPTION

COPIES A FORM FROM THE OPEN LIST TO THE DISPLAY LIST AND
PERFORMS THE
NECESSARY CLEANUP (E.G. ASSIGNING WINDOW IDS, CALCULATING
FIELDS, ETC.).

ARGUMENTS:

EFNAME =	ENAME
PARPTR =	FIELD *
NXTPTR =	FIELD **
PRVPTR =	FIELD **

INCLUDE FILES:

STDTP	- STANDARD TYPE DEFINITIONS
FPD	- FORM PROCESSOR DATA
FPCODE	- FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

ESCPY	- EXTERNAL STRING COPY
STRUPC	
GOFPTR	- GET OPEN FROM POINTER
SYSMSG	- SYSTEM MESSAGE ROUTINE
COPFLD	- COPY FIELD
MAX	
RSVEXP	- RESOLVE EXPRESSIONS
OPNFRM	- OPEN FORM

CALLED DIRECTLY BY:

ADDFRM	-	ADD FORM TO WINDOW
RPLFRM	-	REPLACE FORM
STUPFP	-	SET UP FORM PROCESSOR DATA STRUCTURES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: CURPOS
PURPOSE: GET CURSOR POSITION
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FIELD * ()
SOURCE FILE: CURPOS
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

FIELD *CURPOS()

OUTPUTS:

RETURNS A POINTER TO THE FIELD CONTAINING THE CURSOR.

DESCRIPTION

RETURNS A POINTER TO THE SMALLEST FIELD CONTAINING THE
CURSOR WITH THE
EXCEPTION OF TRANSPARENT FIELDS.

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA

ROUTINES CALLED:

CURPOS/FND CP - FIND CURSOR POSITION

CALLED DIRECTLY BY:

GETCUR - GET CURSOR POSITION

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: CURPOS/FNDCP
PURPOSE: FIND CURSOR POSITION
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FIELD * ()
SOURCE FILE: CURPOS
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

DESCRIPTION

RETURNS A POINTER TO THE SMALLEST FIELD WHICH ENCLOSSES
THE CURSOR.

A FORM OR A WINDOW MUST BE NONTRANSPARENT TO BE
CONSIDERED.

ARGUMENTS:

FLDPTR = FIELD *

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA

ROUTINES CALLED:

ABS
CURPOS/FNDCP - FIND CURSOR POSITION
ACRPOS - ABSOLUTIZE CURSOR POSITION OF FIELD

CALLED DIRECTLY BY:

CURPOS/FNDCP - FIND CURSOR POSITION
CURPOS - GET CURSOR POSITION

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: DBCFNC
PURPOSE: CHECK FUNCTION
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: DBCFNC
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: UIS
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

THIS MODULE CHECKS A FUNCTION TO SEE IF IT IS
A VALID UIMS OR REMOTE APPLICATION.

ARGUMENTS:

CURSOR = RECRD
ROLE = DSPLY [X(10)]
FUNCTION = DSPLY [X(10)]
TYP = DSPLY [X]
RCODE = DSPLY [X(5)]

INCLUDE FILES:

ORACLE - data delcarations for programs that access
ORACLE
ORCODE - ORacle CODEs
CICODE - Command Interpreter CODEs
FPCODE - FORM PROCESSOR RETURN CODES
CURSORI - CURSOR description

ROUTINES CALLED:

OSQL3
OBIND
OEXEC
ODFINN
OFETCH

CALLED DIRECTLY BY:

UIS/STRTAP - START APPLICATION

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: DBCROL
PURPOSE: CHECK ROLE
LANGUAGE: VAX-11 COBOL
MODULE TYPE: SUBROUTINE
SOURCE FILE: DBCROL
SOURCE FILE TYPE: .COB
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: UIS
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

THIS MODULE CHECKS A ROLE TO SEE IF IT IS
VALID FOR A USER ID.

ARGUMENTS:

CURSOR = RECRD
USERID = DSPLY [X(10)]
ROLE = DSPLY [X(10)]
RCODE = DSPLY [X(5)]

INCLUDE FILES:

ORACLE - data declarations for programs that access
ORACLE
ORCODE - Oracle CODES
CICODE - Command Interpreter CODES
FPCODE - FORM PROCESSOR RETURN CODES
CURSORI - CURSOR description

ROUTINES CALLED:

OSQL3
OBIND
OEXEC
ODFINN
OFETCH

CALLED DIRECTLY BY:

UIS - USR INTERFACE SERVICES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: DELFLD
PURPOSE: DELETE FIELD
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: DELFLD
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
CHAR *DELFLD(DP, PDP)  
FIELD *DP, **PDP;
```

INPUTS:

DP - POINTER TO FIELD TO BE DELETED
PDP- POINTER TO POINTER TO THE TOP LEVEL OF THIS LIST.

OUTPUTS:

RETURNS AN ERROR CODE OR NULL (NO ERROR CAN OCCUR
UNLESS DELETING FROM
THE OPEN LIST)

DESCRIPTION

DELETES A FIELD FROM THE OPEN LIST OR THE DISPLAY LIST BY
FIXING UP ALL
THE PERTINENT POINTERS AND FREEING THE

ARGUMENTS:

DP = FIELD *
PDP = FIELD **

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
BITS - INCLUDE FILE FOR BIT MANIPULATION ROUTINES

ROUTINES CALLED:

SYSMSG - SYSTEM MESSAGE ROUTINE
DELFLD - DELETE FIELD

FREE
DEFLD/DELEXP - DELETE EXPRESSION
CBIT

CALLED DIRECTLY BY:

CLSFRM - CLOSE FORM
DEFLD - DELETE FIELD
OPNFRM - OPEN FORM
RMVFPD - REMOVE FORM PROCESSOR DATA STRUCTURE
RMVPAG - REMOVE PAGE
RPLFRM - REPLACE FORM

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: DELFLD/DELEXP
PURPOSE: DELETE EXPRESSION
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: DELFLD
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

STATIC VOID DELEXP(EP)
ENODE *EP;

INPUTS:

EP - POINTER TO EXPRESSION TO DELETE

DESCRIPTION

FREES THE SPECIFIED EXPRESSION TREE.

ARGUMENTS:

EP = ENODE *

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
BITS - INCLUDE FILE FOR BIT MANIPULATION ROUTINES

ROUTINES CALLED:

DELFLD/DELEXP - DELETE EXPRESSION
FREE

CALLED DIRECTLY BY:

DELFLD/DELEXP - DELETE EXPRESSION
DELFLD - DELETE FIELD

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: ESCPY
PURPOSE: EXTERNAL STRING COPY
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: ESCPY
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
VOID ESCPY(TO, FROM, LEN)
    CHAR TO[];
    CHAR FROM[];
    INT LEN;
```

INPUTS:

```
FROM[] - STRING TO COPY FROM
LEN    - MAXIMUM NUMBER OF CHARACTERS TO COPY
```

OUTPUTS:

```
TO - STRING TO COPY INTO
```

DESCRIPTION

ESCPY COPIES "LEN" CHARACTERS FROM "FROM" TO "TO" AND
THEN REMOVES ANY
TRAILING BLANKS FROM "TO".

ARGUMENTS:

```
TO =      CHAR []
FROM =    CHAR []
LEN =     INT
```

INCLUDE FILES:

```
STDYTP - STANDARD TYPE DEFINITIONS
```

ROUTINES CALLED:

```
STRNCPY
```

CALLED DIRECTLY BY:

```
ADDELM - ADD ELEMENT
```

ADDFRM - ADD FORM TO WINDOW
CLSFRM - CLOSE FORM
CMPFLD/EVA - EVALUATE FIELD EXPRESSION
COPFRM - COPY FORM
GDATA - GET DATA
GDATLN - GET DATA LENGTH
GETATT - GET ATTRIBUTE
GETBAK - GET BACKGROUND ATTRIBUTE
GPAGE - GET PAGE
GWINDO - GET WINDOW
MAKAP - MAKE APPLICATION STRUTURE
MAKPD - MAKE PHYSICAL DEVICE STRUCTURE
MONITR/GETPD - GET PHYSICAL DEVICE
MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS
ONWISC - OUTPUT (NO WAIT) / INPUT SCREEN
OPNFRM/PFREC - PROCESS FORM RECORD
OPNFRM/PDREC - PROCESS FIELD RECORD
OPNFRM/PIT - PROCESS ITEM
OPNFRM/PWI - PROCESS WINDOW
OPNFRM - OPEN FORM
PARFQN - PARSE FULLY QUALIFIED NAME
PDATA - PUT FORM DATA
PMSGLS - PUT MESSAGE LINE STRING
PUTATT - PUT ATTRIBUTES
PUTBAK - PUT BACKGROUND ATTRIBUTES
PUTCUR - PUT CURSOR
PUTLOC - PUT LOCATION
RMVPAG - REMOVE PAGE
RPLFRM - REPLACE FORM
SFPDAP - SET FORM PROCESSOR DATA STRUCTURE FOR APLICATION
UIS/STRTAP - START APPLICATION
UIS/STRTPD - START PHYSICAL DEVICE
UIS/FLWNST - FILL WINDOW MANAGER STRUCTURE
UIS/PRCINP - PRCESS INPUT
UIS - USR INTERFACE SERVICES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: FNDFLD
PURPOSE: FIND FIELD
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FIELD * ()
SOURCE FILE: FNDFLD
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

FIELD *FNDFLD(NAME, DP)
NAME NAME;
FIELD *DP;

INPUTS:

NAME - NAME OF THE FIELD TO FIND
DP - POINTER TO FORM TO LOOK FOR FIELD

OUTPUTS:

RETURNS POINTER TO FIELD

DESCRIPTION

GIVEN THE NAME OF A FIELD AND THE FORM IT IS ON FNDFLD
RETURNS A
POINTER TO THE FIELD.

ARGUMENTS:

NAME = NAME
DP = FIELD *

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA

ROUTINES CALLED:

STRCMP

CALLED DIRECTLY BY:

OPNFRM/BRPNOD - BUILD RELATIVE POSITION NODE

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: FNDMSG
PURPOSE: FIND MESSAGE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: FNDMSG
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
VOID FNDMSG(CODEP, MSGSTR)
    CHAR *CODEP;
    MSG MSGSTR;
```

INPUTS:

CODEP - RETURN CODE TO FIND MESSAGE FOR

OUTPUTS:

MSGSTR - MESSAGE CORRESPONDING TO CODEP

DESCRIPTION

FIND THE MESSAGE STRING THAT CORRESPONDS TO THE CODE.

ARGUMENTS:

```
CODEP = CHAR *
MSGSTR = MSG
```

INCLUDE FILES:

```
STDTPY - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FPMSG - FORM PROCESSOR ERROR MESSAGES
```

ROUTINES CALLED:

```
STRNCMP
FNDMSG/OUNSGF - OPEN USER MESSAGE FILE
FNDMSG/CODSCH - CODE SEARCH
SPRINTF
STRLEN
MEMSET
```

CALLED DIRECTLY BY:

PMSGC - PUT MESSAGE LINE CODE
SYSMSG - SYSTEM MESSAGE ROUTINE

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: FNDMSG/CODSCH
PURPOSE: CODE SEARCH
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: FNDMSG
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

ARGUMENTS:

CODEP = CHAR *
FP = FILE *
MSGSTR = CHAR *

INCLUDE FILES:

STDTPY - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FPEMSG - FORM PROCESSOR ERROR MESSAGES

ROUTINES CALLED:

MEMCPY
MEMCMP
FREAD
REWIND

CALLED DIRECTLY BY:

FNDMSG - FIND MESSAGE

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: FNDMSG/OUMSGF
PURPOSE: OPEN USER MESSAGE FILE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: FNDMSG
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
STATIC CHAR *OUMSGF(CODEP)
CHAR CODEP[];
```

INPUTS:

CODEP - RETURN CODE TO OPEN MESSAGE FILE FOR

DESCRIPTION

OUMSGF OPENS THE MESSAGE FILE CORRESPONDING TO THE
SPECIFIED RETURN CODE.

ARGUMENTS:

CODEP = CHAR []

INCLUDE FILES:

```
STDTyp - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FPEMSG - FORM PROCESSOR ERROR MESSAGES
```

ROUTINES CALLED:

MEMCPY
FOPEN
SPRINTF

CALLED DIRECTLY BY:

FNDMSG - FIND MESSAGE

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: FNNPWN
PURPOSE: FIND FORM PROCESSOR WINDOW
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FIELD * ()
SOURCE FILE: FNNPWN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

FIELD *FNNPWN(FLDPT, WNDID)
FIELD *FLDPT;
INT WNDID;

INPUTS:

FLDPT - POINTER TO FIRST WINDOW IN LIST TO BE SEARCHED
WNDID - ID OF WINDOW SEARCHING FOR

OUTPUTS:

RETURNS A POINTER TO WINDOW FOUND OR A NULL

DESCRIPTION

THIS MODULE SEARCHES FOR A WINDOW WITH THE ID GIVEN AND
EITHER RETURNS
A POINTER TO THE WINDOW FOUND OR A NULL.

ARGUMENTS:

FLDPT = FIELD *
WNDID = INT

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA

ROUTINES CALLED:

FNNPWN - FIND FORM PROCESSOR WINDOW

CALLED DIRECTLY BY:

CLSLDV - CLOSE LOGICAL DEVICE
FNNPWN - FIND FORM PROCESSOR WINDOW

GDVINP - GET DEVICE INPUT
OISCR/EVTBUF - EMPTY VTI BUFFER
UIS/PRCINP - PROCESS INPUT

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: FUISWN
PURPOSE: FIND UIS WINDOW
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FPD * ()
SOURCE FILE: FUISWN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

FPD *FUISWN(PDPTR)
PD *PDPTR;

INPUTS/OUTPUTS:

INPUTS:

PDPTR - POINTER TO PHYSICAL DEVICE ON WHICH TO LOOK FOR
UIS WINDOW

OUTPUTS:

RETURNS A POINTER TO UIS LOG DEV IF FOUND UIS OTHERWISE
RETURNS NULL

DESCRIPTION

THIS MODULE SEARCHES FOR UIS WINDOW ON SPECIFIED PHYSICAL
DEVICE.
IT RETURNS EITHER A POINTER TO THE UIS LOG DEV (IF FOUND)
OR A NULL
POINTER.

ARGUMENTS:

PDPTR = PD *

INCLUDE FILES:

STDTPY - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA

CALLED DIRECTLY BY:

INSCR - INPUT SCREEN
PMSGLS - PUT MESSAGE LINE STRING

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: GARPOS
PURPOSE: GET ARRAY OFFSET POSITION OF FIELD
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: GARPOS
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION: -----

SYNOPSIS

```
VOID GARPOS(DP, PARPTR, ARYPOS)
    FIELD      *DP;
    FIELD      **PARPTR;
    POSITION    *ARYPOS;
```

INPUTS/OUTPUTS:

INPUTS:

DP - FIELD WHOSE OFFSET ROW AND COL FROM FIRST ARRAY
DIMMENSION WANTED
ADDRESS OF STURCTURE FOR RETURNING VALUES OF:

OFFSET ROW
OFFSET COL

OUTPUTS:

ARYPTR - FIRST ARRAY DIMMENSION INTERESTED IN
STRUCTURE CONTAINING:
OFFSET ROW OF FIELD
OFFSET COL OF FIELD

DESCRIPTION

THIS MODULE OBTAINS A THE OFFSET ROW AND COL OF A FIELD
FROM FIRST
DIMMENSION OF AN ARRAY BY GOING BACK UP CHILD PARENT TREE
AND ADDING
EACH SUCCESSIVE PARENT'S ROW AND COL TO SUM OF CHILDS'
FOR ALL
DIMMENSIONS OF THE ARRAY.

ARGUMENTS: -----

DP = FIELD *
PARPTR = FIELD **
ARYPOS = POSITION *

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA

FORM PROCESSOR Module Documentation

NAME: GATDEF
PURPOSE: GET ATTRIBUTE DEFINITION
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: GATDEF
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
CHAR *GATDEF(ATNAME, ADP)
    NAME    ATNAME;
    ATTMAP **ADP;
```

INPUTS:

ATNAME - NAME OF ATTRIBUTE

OUTPUTS:

ADP - POINTER TO ATTRIBUTE MAP OF AN ATTRIBUTE NAME
AND
DEFINITION.

RETURNS ERROR CODE OR NULL IF SUCCESSFUL.

DESCRIPTION

TRANSLATES ATTRIBUTE NAME INTO A BIT MAP OF AN ATTRIBUTE
DEFINITION

ARGUMENTS:

ATNAME = NAME
ADP = ATTMAP **

INCLUDE FILES:

STDTPY - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

STRUPC
STRCMP
SYSMSG - SYSTEM MESSAGE ROUTINE

CALLED DIRECTLY BY:

ONWISC - OUTPUT (NO WAIT) / INPUT SCREEN
OPNFRM/PFREC - PROCESS FORM RECORD
OPNFRM/PAR - PROCESS ARRAY
OPNFRM/PIT - PROCESS ITEM
OPNFRM/PWI - PROCESS WINDOW
PUTATT - PUT ATTRIBUTES
PUTBAK - PUT BACKGROUND ATTRIBUTES
STUPFP - SET UP FORM PROCESSOR DATA STRUCTURES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: GDATA
PURPOSE: GET DATA
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FORTRAN VOID ()
SOURCE FILE: GDATA
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
FORTRAN VOID GDATA(INSTID, EWPATH, FDATA, CODEP)
  INT      *INSTID;
  EPATH    EWPATH;
  CHAR     *FDATA;
  CHAR     CODEP[];
```

INPUTS:

INSTID - INSTANCE ID (PREV OR CURRNT)
EWPATH - PATH NAME

OUTPUTS:

FDATA - REQUESTED DATA
CODEP - RETURN CODE

DESCRIPTION

GDATA IS USED TO GET USER ENTERED DATA. THE DATA CAN BE FROM A FORM, WINDOW, ARRAY, OR FIELD DEPENDING ON THE PATH (EWPATH).

ARGUMENTS:

INSTID = INT *
EWPATH = EPATH
FDATA = CHAR *
CODEP = CHAR []

INCLUDE FILES:

STDTPY - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS

ROUTINES CALLED:

ESCPY - EXTERNAL STRING COPY
PTHPTR - GET PATH POINTER
GDATA/GETBUF - GET BUFFER
SYSMSG - SYSTEM MESSAGE ROUTINE
MEMCPY

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES
UIS - USR INTERFACE SERVICES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: GDATA/GETBUF
PURPOSE: GET BUFFER
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: GDATA
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

DESCRIPTION

GETBUF PUTS THE CONTENTS OF ALL FIELDS CONTAINED IN THE
DATA AREA
SPECIFIED BY 'DP' INTO THE BUFFER FDATA. IT CALLS ITSELF
TO GET
THE CONTENTS OF SUBAREAS (FORMS, ITEMS, ETC.) WITHIN THE
SPECIFIED
AREA.

ARGUMENTS:

DP = FIELD *

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS

ROUTINES CALLED:

GDATA/GETBUF - GET BUFFER
BLEN
CBPTR
MEMCPY

CALLED DIRECTLY BY:

GDATA/GETBUF - GET BUFFER
GDATA - GET DATA

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: GDATLN
PURPOSE: GET DATA LENGTH
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FORTRAN VOID ()
SOURCE FILE: GDATLN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
FORTRAN VOID GDATLN(EWPATH, LEN, CODEP)
    EPATH  EWPATH;
    INT    *LEN;
    CHAR    CODEP[];
```

INPUTS:

EWPATH - PATH NAME

OUTPUTS:

LEN - LENGTH OF DATA
CODEP - RETURN CODE

DESCRIPTION

GDATLN GETS THE LENGTH OF THE DATA CORRESPONDING TO THE
FORM, WINDOW,
OR ARRAY SPECIFIED BY THE PATH.

ARGUMENTS:

EWPATH = EPATH
LEN = INT *
CODEP = CHAR []

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS

ROUTINES CALLED:

ESCPY - EXTERNAL STRING COPY

PTHPTR - GET PATH POINTER
GDATLN/GBUFLN - GET BUFFER LENGTH
MEMCPY

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: GDATLN/GBUFLN
PURPOSE: GET BUFFER LENGTH
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: GDATLN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

DESCRIPTION
RETURNS THE LENGTH OF THE SPECIFIED BUFFER

ARGUMENTS:

DP = FIELD *

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS

ROUTINES CALLED:

GDATLN/GBUFLN - GET BUFFER LENGTH
BLEN

CALLED DIRECTLY BY:

GDATLN/GBUFLN - GET BUFFER LENGTH
GDATLN - GET DATA LENGTH

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: GDVINP
PURPOSE: GET DEVICE INPUT
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: GDVINP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
CHAR *GDVINP(PDPTR, MSGBUF, LEN, APNAM, APCHAN)
  PD *PDPTR;
  CHAR **MSGBUF;
  INT *LEN;
  NAME APNAM;
  CHAN APCHAN;
```

INPUTS:

PDPTR - POINTER TO USER'S PHYSICAL DEVICE
MSGBUF - CONTAINS DATA FROM DEVICE DRIVER
LEN - LENGTH OF MSGBUF

OUTPUTS:

APNAM - NAME OF AP STRUCTURE WILL BE FOR - FROM NTM
APCHAN - UNIQUE INSTANCE OF AP - FROM NTM
RETURNS STANDARD RETURN CODE FOR FORM PROCESSOR

DESCRIPTION

THIS MODULE IS CALLED BY THE MONITOR TO SETUP THE FPD
STRUCTURE
AND GET INPUT FROM THE DEVICE DRIVER RETURNING THE
APPLICATION NAME
AND CHANNEL OF APPLICATION FOR WHICH THE DATA WAS FOR. IF
THE RCODE
RETURNED IS INPNCMPL INPUT WAS NOT COMPLETED.

ARGUMENTS:

PDPTR =	PD *
MSGBUF =	CHAR **
LEN =	INT *
APNAM =	NAME
APCHAN =	CHAN

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FUNCTS - FUNCTION DEFINITIONS
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

BLDCMD
MEMSET
PUTW
FWRITE
SYSMSG - SYSTEM MESSAGE ROUTINE
FNFPWN - FIND FORM PROCESSOR WINDOW
PDVOTP - PUT DEVICE OUTPUT
FTELL
FCLOSE
MEMCPY
STRLEN
INSCR - INPUT SCREEN
GVTICMD

CALLED DIRECTLY BY:

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: GETATT
PURPOSE: GET ATTRIBUTE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FORTRAN VOID ()
SOURCE FILE: GETATT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
FORTRAN VOID GETATT(EWPATH, DUR, ATTRBT, RCODE)
    EPATH  EWPATH;
    INT    *DUR;
    ENAME  ATTRBT;
    CHAR   RCODE[];
```

INPUTS:

EWPATH - QUALIFIED NAME OF FIELD OF WHICH CALLER
 WISHES ATTRIBUTES
DUR - PERM/TEMP FLAG

OUTPUTS:

ATTRBT - NAME OF ATTRIBUTE
RCODE - RETURN CODE

DESCRIPTION

GETATT GETS THE ATTRIBUTE IDENTIFIERS FOR ANY ITEM FIELD.

ARGUMENTS:

EWPATH =	EPATH
DUR =	INT *
ATTRBT =	ENAME
RCODE =	CHAR []

INCLUDE FILES:

STDTP	- STANDARD TYPE DEFINITIONS
FPD	- FORM PROCESSOR DATA
FPCODE	- FORM PROCESSOR RETURN CODES
FPPARM	- FORM PROCESSOR PARAMETERS

ROUTINES CALLED:

ESCPY - EXTERNAL STRING COPY

PTHPTR - GET PATH POINTER
SYMSG - SYSTEM MESSAGE ROUTINE
MEMSET
STRLEN
MEMCPY

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: GETBAK
PURPOSE: GET BACKGROUND ATTRIBUTE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FORTRAN VOID ()
SOURCE FILE: GETBAK
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
FORTRAN VOID GETATT(EWPATH, DUR, ATTRBT, RCODE)
    EPATH  EWPATH;
    INT    *DUR;
    ENAME  ATTRBT;
    CHAR   RCODE[];
```

INPUTS:

EWPATH = QUALIFIED NAME OF FEILD OF WHICH CALLER
 WISHES ATTRIBUTES
DUR = PERM/TEMP FLAG

OUTPUTS:

ATTRBT = NAME OF ATTRIBUTE
RCODE = RETURN CODE

DESCRIPTION

GETBAK GETS THE ATTRIBUTE IDENTIFIERS FOR WINDOW AND FORMS.

ARGUMENTS:

EWPATH = EPATH
DUR = INT *
ATTRBT = ENAME
RCODE = CHAR []

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS

ROUTINES CALLED:

ESCPY - EXTERNAL STRING COPY

PTHPTR - GET PATH POINTER
SYMSG - SYSTEM MESSAGE ROUTINE
MEMSET
STRLEN
MEMCPY

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: GETCUR
PURPOSE: GET CURSOR POSITION
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FORTRAN VOID ()
SOURCE FILE: GETCUR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

FORTRAN VOID GETCUR(FQ_NAM,TYPE, ROW, COL, RCODE)

EPATH FQ_NAM;
CHAR *TYPE;
INT *ROW;
INT *COL;
CHAR RCODE[];

OUTPUTS:

FQ_NAM = FULLY QUALIFIED NAME OF FIELD IN WHICH CURSOR LIES
- INITIALIZED TO SPACES
ARRAYS - DIMENSIONS IN WHICH CURSOR FOUND INCLUDED
WINDOWS - PAGE ON WHICH CURSOR FOUND INCLUDED
TYPE = TYPE OF FIELD IN WHICH CURSOR FOUND
- INITIALIZED TO SPACES
ROW = ROW WITHIN FIELD - INITIALIZED TO ABSOLUTE ROW ON
SCREEN
COL = COL WITHIN FIELD - INITIALIZED TO ABSOLUTE COL ON
SCREEN
RCODE = RETURNS ERROR IF FULLY QUALIFIED NAME TOO
LONG

DESCRIPTION

THIS ROUTINE RETURNS THE FULLY QUALIFIED NAME OF THE
FIELD IN
WHICH THE CURSOR IS FOUND [THE DIMENSIONS (IF ANY) OF
ARRAY IN WHICH
IT LIES AS WELL AS PAGE NUMBER IS INCLUDED IN FULLY
QUALIFIED NAME]
AND ROW AND COLUMN WITHIN THIS FIELD

ARGUMENTS:

FQ_NAM = EPATH
TYPE = CHAR *

ROW = INT *
COL = INT *
RCODE = CHAR []

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS

ROUTINES CALLED:

MALLOC
SPRINTF
GETCUR/CONCAT - CONCATENATE STRING TO CURRENT NAME
ACRPOS - ABSOLUTIZE CURSOR POSITION OF FIELD
MEMSET
FREE
SYSMSG - SYSTEM MESSAGE ROUTINE
MEMCPY
CURPOS - GET CURSOR POSITION

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: GETCUR/CONCAT
PURPOSE: CONCATENATE STRING TO CURRENT NAME
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: GETCUR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

ARGUMENTS:

STR = CHAR []

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS

ROUTINES CALLED:

MEMCPY
STRLEN

CALLED DIRECTLY BY:

GETCUR - GET CURSOR POSITION

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: GOFPTR
PURPOSE: GET OPEN FROM POINTER
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FIELD * ()
SOURCE FILE: GOFPTR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

FIELD *GOFPTR(FNAME)
NAME FNAME;

INPUTS:

FNAME - NAME OF FORM TO GET POINTER TO

OUTPUTS:

RETURNS POINTER TO FORM ON OPEN LIST

DESCRIPTION

RETURNS A POINTER TO THE SPECIFIED FORM ON THE OPEN LIST
OR NULL IF
NOT FOUND.

ARGUMENTS:

FNAME = NAME

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA

ROUTINES CALLED:

STRCMP

CALLED DIRECTLY BY:

ADDELM - ADD ELEMENT
CLSFRM - CLOSE FORM
COPFRM - COPY FORM
OPNFRM/PFR - PROCESS FORM
OPNFRM - OPEN FORM

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: GPAGE
PURPOSE: GET PAGE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FORTRAN VOID ()
SOURCE FILE: GPAGE
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
FORTRAN VOID GPAGE(EWPATH, PNUMP, EFNAME, CODEP)
    EPATH  EWPATH;
    INT    *PNUMP;
    ENAME   *EFNAME;
    CHAR    CODEP[];
```

INPUTS:

EWPATH - PATH NAME OF WINDOW
PNUMP - PAGE NUMBER IN WINDOW

OUTPUTS:

EFNAME - FORM NAME
CODEP - RETURN CODE

DESCRIPTION

RETURNS THE NAME OF THE FORM ON THE SPECIFIED PAGE IN THE WINDOW.

ARGUMENTS:

EWPATH = EPATH
PNUMP = INT *
EFNAME = ENAME
CODEP = CHAR []

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS

ROUTINES CALLED:

ESCPY - EXTERNAL STRING COPY

PTHPTR - GET PATH POINTER
SYSMSG - SYSTEM MESSAGE ROUTINE
MEMCPY
STRLEN
MEMSET

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: GWINDO
PURPOSE: GET WINDOW
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FORTRAN VOID ()
SOURCE FILE: GWINDO
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
FORTRAN VOID GWINDO(EWPATH, NUMPP, CODEP)
    EPATH  EWPATH;
    INT    *NUMPP;
    CHAR    CODEP[];
```

INPUTS:

EWPATH - PATH NAME OF WINDOW

OUTPUTS:

NUMPP - NUMBER OF PAGES IN WINDOW
CODEP - RETURN CODE

DESCRIPTION

GETS THE NUMBER OF PAGES IN A WINDOW.

ARGUMENTS:

EWPATH =	EPATH
NUMPP =	INT *
CODEP =	CHAR []

INCLUDE FILES:

STDTP	- STANDARD TYPE DEFINITIONS
FPD	- FORM PROCESSOR DATA
FPCODE	- FORM PROCESSOR RETURN CODES
FPPARM	- FORM PROCESSOR PARAMETERS

ROUTINES CALLED:

ESCPY	- EXTERNAL STRING COPY
PTHPTR	- GET PATH POINTER
MEMCPY	

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: INITVT
PURPOSE: INITIAL VIRTUAL TERMINAL INTERFACE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: INITVT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

INITVT(RCODE)
CHAR RCODE[]

INPUTS/OUTPUTS:

INPUTS:
NONE

OUTPUTS:
RCODE - STANDARD FORM PROCESSOR RETURN CODE

DESCRIPTION

SET VTI MODE FLAG AND INITIALIALIZE MAX BUFFER LENGTH TO 0

ARGUMENTS:

RCODE = CHAR []

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

MEMCPY

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: INQLDV
PURPOSE: INQUIRE LOGICAL DEVICE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FORTRAN VOID ()
SOURCE FILE: INQLDV
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
FORTRAN VOID INQLDV(LDWNID,RCODE)
      INT          *LDWNID;
      CHAR          RCODE[];
```

INPUTS/OUTPUTS:

INPUTS:

ADDRESS OF: LOGICAL DEVICES TOP WINDOW ID
RETURN CODE

OUTPUTS:

LDWNID - LOGICAL DEVICES TOP WINDOW ID
RCODE - RETURN CODE

DESCRIPTION

THIS MODULE RETURNS LOGICAL DEVICE ON WHICH APPLICATION
CURRENTLY RUNNING.

IF NO LOGICAL DEVICE FOUND WILL RETURN NFPDSTRC ERROR.

ARGUMENTS:

```
LDWNID = INT *
RCODE = CHAR [ ]
```

INCLUDE FILES:

```
STD TYP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
```


ROUTINES CALLED:

MEMCPY

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: INSCR
PURPOSE: INPUT SCREEN
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: OISCR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

----- SYNOPSIS

```
CHAR *INSCR(PDPTR, WPTR, VBPTR, VBEND)
    PD      *PDPTR;
    FIELD   *WPTR;
    CHAR     *VBPTR, *VBEND;
```

INPUTS:

PDPTR - POINTER TO PHYSICAL DEVICE INPUT IS FROM
WPTR - POINTER TO WINDOW INPUT IS FOR
VBPTR - POINTER TO BEGINNING OF INPUT DATA
VBEND - POINTER TO (CHARACTER PAST) END OF INPUT DATA

OUTPUTS:

RETURNS A RETURN CODE

DESCRIPTION

PROCESS THE VIRTUAL TERMINAL INPUT FOR THE SPECIFIED WINDOW.

ARGUMENTS:

PDPTR = PD *
WPTR = FIELD *
VBPTR = CHAR *
VBEND = CHAR *

INCLUDE FILES:

STDITYP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
TIME - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS
VTICOM - VTI COMMUNICATION DEFINITIONS
FUNCTS - FUNCTION DEFINITIONS
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

OISCR/EVTBUF - EMPTY VTI BUFFER
FUISWN - FIND UIS WINDOW
SYSMSG - SYSTEM MESSAGE ROUTINE
FREMSG
OISCR/RSTMAT - RESET TEMPORAY ATTRIBUTES
PFINP
MEMCMP
PMSGLC - PUT MESSAGE LINE CODE
OISCR/FVTBUF - FILL VTI BUFFER
DSPMSG

CALLED DIRECTLY BY:

GDVINP - GET DEVICE INPUT

FORM PROCESSOR Module Documentation

NAME: MABSAT
PURPOSE: MAP ABSOLUTE ATTRIBUTE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: MABSAT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
VOID MABSAT(BADP, ATTRIB, FADP)
    ATTDEF *BADP;
    ATTMAP *ATTRIB;
    ATTDEF *FADP;
```

INPUTS:

BADP = BACKGROUND ATTRIBUTE (ABSOLUTE)
ATTRIB = FOREGROUND ATTRIBUTE TO COMBINE WITH BACKGROUND

OUTPUTS:

FADP = FOREGROUND ATTRIBUTE (ABSOLUTE)

DESCRIPTION

BUILD AN ATTRIBUTE DEFINITION BASED ON A BACKGROUND
ATTRIBUTE
DEFINITION AND AN ATTRIBUTE NAME.

ARGUMENTS:

BADP = ATTDEF *
ATTRIB = ATTMAP *
FADP = ATTDEF *

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA

ROUTINES CALLED:

STRASN

CALLED DIRECTLY BY:

OISCR/GATINF - GET ATTRIBUTE INFO
PUTATT/AABSAT - ATTRIBUTE ABSOLUTE SET ATTRIBUTE
RSVATT/RSVRST - RESOLVE REST

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: MAKAP
PURPOSE: MAKE APPLICATION STRUCTURE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: MAKAP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
CHAR *MAKAP(PDPTR, APNAM, APCHAN)
PD *PDPTR
NAME APNAM;
CHAN APCHAN;
```

INPUTS/OUTPUTS:

INPUTS:

PDPTR - POINTER TO PHYSICAL DEVICE STRUCTURE
APNAM - NAME OF AP STRUCTURE WILL BE FOR - FROM NTM
APCHAN - UNIQUE INSTANCE OF AP - FROM NTM

OUTPUTS:

RETURNS NULL IF SUCCESSFUL ELSE RETURNS ADDRESS OF
ERROR CODE

DESCRIPTION

THIS MODULE CREATES AND INSERTS A AP DATA STRUCTURE FOR A
USER

ARGUMENTS:

PDPTR =	PD *
APNAM =	NAME
APCHAN =	CHAN

INCLUDE FILES:

STDTP	- STANDARD TYPE DEFINITIONS
FPD	- FORM PROCESSOR DATA
FPCODE	- FORM PROCESSOR RETURN CODES

FORM PROCESSOR Module Documentation

NAME: MAKFLD
PURPOSE: MAKE FIELD
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * (
SOURCE FILE: MAKFLD
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
CHAR *MAKFLD(PDP, RDP, LDP, PP, FLDNAM, ROW, COL, WIDTH,
             DEPTH, TYPE,
             PERMAT)
REGISTER FIELD **PDP;
FIELD **RDP, **LDP, *PP;
NAME FLDNAM;
INT ROW, COL, WIDTH, DEPTH;
CHAR TYPE;
ATTMAP *PERMAT;
```

INPUTS/OUTPUTS:

FIELD **PDP - POINTER TO A POINTER TO BE SET TO THE
NEWLY CREATED FIELD

INPUTS:

FIELD **RDP - POINTER TO SET *PDP'S NXTFLD TO: EITHER A
&CONPTR, OR
&NXTFLD. *RDP IS THE VALUE YOU WANT TO
SET *PDP->NXTFLD
TO AND THE POINTER YOU WANT SET TO *PDP.

FIELD **LDP - POINTER TO SET *PDP'S PRVFLD TO: EITHER A
&LSTPTR, OR
&PRVFLD (OR NULL IF THERE IS NONE). *LDP
IS THE VALUE YOU
WANT TO SET *PDP->PRVFLD TO AND THE
POINTER YOU WANT TO
*PDP. IF LDP IS NULL *PDP->PRVFLD IS SET
TO NULL ONLY.

FIELD *PP - PARENT OF NEWLY CREATED FIELD.

NAME FLDNAM - NULL TERMINATED STRING WITH NAME OF NEW
FIELD.

INT ROW, COL, WIDTH, DEPTH - FOR THE NEW FIELD,
ABSOLUTE POSITIONS.

CHAR TYPE - FIELD TYPE (A, F, I, W).

ATTMAP *PERMAT - ATTRIBUTE MAP ELEMENT TO PUT IN THIS
FIELD.

DESCRIPTION

MAKFLD CREATES A FIELD, THE FIELD'S POINTER IS RETURNED
IN PDP,
THE FIELD'S PARENT IS POINTED TO BY PP, THE POSITION TO
CHAIN
IN THE FIELD IS POINTED TO BY RDP AND LDP.

ARGUMENTS:

PDP = FIELD **
RDP = FIELD **
LDP = FIELD **
PP = FIELD *
FLDNAM = NAME
ROW = INT
COL = INT
WIDTH = INT
DEPTH = INT
TYPE = CHAR
PERMAT = ATTMAT *

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

MALLOC
SYSMSG - SYSTEM MESSAGE ROUTINE
STRCPY

CALLED DIRECTLY BY:

OPNFRM/PFREC - PROCESS FORM RECORD
OPNFRM/PAR - PROCESS ARRAY
OPNFRM/PIT - PROCESS ITEM
OPNFRM/PFR - PROCESS FORM
OPNFRM/PWI - PROCESS WINDOW
STUPFP - SET UP FORM PROCESSOR DATA STRUCTURES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: MAKFPD
PURPOSE: MAKE FORM PROCESSOR DATA (LOGICAL DEVICE
STRUCTURE)
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FPD * ()
SOURCE FILE: MAKFPD
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
FPD *MAKFPD(PDPTR, APPTR)
  PD *PDPTR;
  AP *APPTR;
```

INPUTS/OUTPUTS:

INPUTS:

PDPTR - POINTER TO PHYSICAL DEVICE IT WILL BELONG TO
APPTR - POINTER TO APPLICATION IT WILL BELONG TO

OUTPUTS:

RETURNS A POINTER TO STRUCTURE CREATED IF SUCCESSFUL
ELSE NULL POINTER

DESCRIPTION

THIS MODULE CREATES A LOGICAL DEVICE STRUCTURE FOR AN
APPLICATION ON
A PARTICULAR PHYSICAL DEVICE.

ARGUMENTS:

```
PDPTR = PD *
APPTR = AP *
```

INCLUDE FILES:

```
STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
```

ROUTINES CALLED:

CALLOC

CALLED DIRECTLY BY:

MAKAP - MAKE APPLICATION STRUTURE
MAKUSR - MAKE USER
OPNLDV - OPEN LOGICAL DEVICE

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: MAKPD
PURPOSE: MAKE PHYSICAL DEVICE STRUCTURE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: PD * ()
SOURCE FILE: MAKPD
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
PD *MAKPD(USRPTR, PDPRT, PDNAM, PDCHAN, SIZ)
    USR      *USRPTR;
    NAME     PDPRT;
    NAME     PDNAM;
    CHAN     PDCHAN;
    SIZE     *SIZ;
```

INPUTS/OUTPUTS:

INPUTS:

USRPTR - POINTER TO USER FOR WHICH PHYSICAL DEVICE
STRUCTURE IS BEING
CREATED.
PDPRT - PHYSICAL PORT OF DEVICE
PDNAM - NAME/TYPE OF DEVICE
PDCHAN - CHAN OF DEVICE - FROM NTM
SIZ - CONTAINS: ACTUAL MAX WIDTH OF DEVICE
ACTUAL MAX DEPTH OF DEVICE

OUTPUTS:

RETURNS A POINTER TO PD STRUCTURE IF SUCCESSFUL AND A
NULL POINTER
IF FAILURE

DESCRIPTION

THIS MODULE CREATES AND INSERTS A STRUCTURE FOR A
PHYSICAL DEVICE FOR
A USER.

ARGUMENTS:

```
USRPTR =      USR *
PDPRT =      NAME
PDNAM =      NAME
PDCHAN =      CHAN
SIZ =      SIZE *
```

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

CALLOC
ESCPY - EXTERNAL STRING COPY

CALLED DIRECTLY BY:

MAKUSR - MAKE USER
UIS/STRTPD - START PHYSICAL DEVICE

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: MAKUSR
PURPOSE: MAKE USER
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: MAKUSR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
CHAR *MAKUSR(PDNAM, PDCHAN, MSGBUF, LEN)
    NAME PDNAM;
    CHAN PDCHAN;
    CHAR *MSGBUF;
    INT *LEN;
```

INPUTS/OUTPUTS:

INPUTS:

PDNAM - NAME OF THE USER'S PHYSICAL DEVICE - FROM NTM
PDCHAN - CHAN OF THE USER'S PHYSICAL DEVICE - FROM NTM
MSGBUF - CONTAINS :PDWIDTH - MAX WIDTH OF PHYSICAL DEVICE
 PDDPTH - MAX DEPTH OF PHYSICAL DEVICE
LEN - LENGTH OF MESSAGE BUFFER

OUTPUTS:

IF ERROR RETURNS POINTER TO ERROR CODE ELSE RETURNS A
NULL POINTER

DESCRIPTION

THIS MODULE IS CALLED BY THE MONITOR TO CREATE AND INSERT
A USER
STRUCTURE FOR A USER; IT CALLS MAKPD TO CREATE AND INSERT
A PHYSICAL
DEVICE FOR THE USER ALSO.

ARGUMENTS:

PDNAM = NAME
PDCHAN = CHAN
MSGBUF = CHAR *
LEN = INT *

INCLUDE FILES:

```

STDTPY      - STANDARD TYPE DEFINITIONS
FPD         - FORM PROCESSOR DATA
FPCODE     - FORM PROCESSOR RETURN CODES
FUNCTS     - FUNCTION DEFINITIONS
CTLCHR     - CONTROL CHARACTERS

```

ROUTINES CALLED:

```

BLDCMD
GVTICMD
SYMSMSG      - SYSTEM MESSAGE ROUTINE
STUPFP       - SET UP FORM PROCESSOR DATA STRUCTURES
TRMUSR       - TERMINATE USER
SPRINTF
PDVOTP       - PUT DEVICE OUTPUT
STRLEN
MAKFPD       - MAKE FORM PROCESSOR DATA (LOGICAL DEVICE
                STRUCTURE)
RMVPD        - REMOVE PHYSICAL DEVICE DATA STRUCTURE
MAKPD        - MAKE PHYSICAL DEVICE STRUCTURE
CFREE
CALLOC

```

CALLED DIRECTLY BY:

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

USED IN MAIN PROGRAM(S) :

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: MONITR/GETPD
PURPOSE: GET PHYSICAL DEVICE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: PD * ()
SOURCE FILE: MONITR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
STATIC PD *GETPD(PDNAM, PDCHAN)
    ENAME PDNAM;
    CHAN PDCHAN;
```

INPUTS:

PDNAM - NAME OF THE USER'S PHYSICAL DEVICE
PDCHAN - CHAN OF THE USER'S PHYSICAL DEVICE

OUTPUTS:

RETURNS POINTER TO THE SPECIFIED PHYSICAL DEVICE
STRUCTURE

DESCRIPTION

SEARCHES THROUGH THE PHYSICAL DEVICE STRUCTURES FOR ONE
WITH THE
SPECIFIED NAME AND CHANNEL. IF NOT FOUND, A NULL POINTER
IS RETURNED.

ARGUMENTS:

PDNAM = ENAME
PDCHAN = CHAN

INCLUDE FILES:

STDTPY - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPDINI - FPD INITIALIZATION
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS
NTM - NTM INTERFACE INCLUDE FILE
DBASEI - DATABASE INTERFACE
FUNCTS - FUNCTION DEFINITIONS
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

STRCMP
ESCPY - EXTERNAL STRING COPY

CALLED DIRECTLY BY:

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: MONITR/MAIN
PURPOSE: MAIN MODULE FOR MONITOR/UIS/FP PROCESS
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: MONITR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS
VOID MAIN()

DESCRIPTION
THIS MODULE IS THE MAIN MODULE FOR MONITOR/UIS/FP
PROCESS. IT
MONITORS INPUT FROM NTM: DECIDING WHAT ACTION IS
REQUIRED AND CALLING
THE APPROPRIATE ROUTINE(S).

INCLUDE FILES:

STDYTP	- STANDARD TYPE DEFINITIONS
STDIO	- **** PURPOSE NOT FOUND BY STRIPPER ****
FPD	- FORM PROCESSOR DATA
FPDINI	- FPD INITIALIZATION
FPCODE	- FORM PROCESSOR RETURN CODES
FPPARM	- FORM PROCESSOR PARAMETERS
NTM	- NTM INTERFACE INCLUDE FILE
DBASEI	- DATABASE INTERFACE
FUNCTS	- FUNCTION DEFINITIONS
CTLCHR	- CONTROL CHARACTERS

ROUTINES CALLED:

BLDCMD	
INITAL	
MEMCMP	
UIS	- USR INTERFACE SERVICES
ESCPY	- EXTERNAL STRING COPY
SFPDAP	- SET FORM PROCESSOR DATA STRUCTURE FOR APPLICATION
RMVAP	- REMOVE APPLICATION
CALLFP	- CALL FP ROUTINES
PUTW	
RMVPD	- REMOVE PHYSICAL DEVICE DATA STRUCTURE
TRMUSR	- TERMINATE USER

STRCHR
MAKUSR - MAKE USER
STRCMP
GVTICMD
MAX
ONWISC - OUTPUT (NO WAIT) / INPUT SCREEN
OUTSCR - OUTPUT SCREEN
FSEEK
PMSGLC - PUT MESSAGE LINE CODE
SYSMSG - SYSTEM MESSAGE ROUTINE
GDVINP - GET DEVICE INPUT
FCLOSE
MONITR/GETPD - GET PHYSICAL DEVICE
FEOF
FERROR
FREAD
GETW
FTELL
FPRINTF
MEMCPY
FWRITE
MIN
SPRINTF
NSEND
PUTC
TRMNAT
MEMSET
RCV
FOPEN
DBCLSE
DBOPEN

FORM PROCESSOR Module Documentation

NAME: OISCR/ADDCMD
PURPOSE: ADD COMMAND TO BUFFER
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: OISCR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

----- SYNOPSIS

```
VOID ADDCMD(D, S, L)
    PD *D;
    CHAR *S;
    INT L;
```

INPUTS:

D - DEVICE SENDING DATA TO
S - STRING TO ADD TO COMMAND BUFFER
L - LENGTH OF STRING

DESCRIPTION

THE SPECIFIED STRING IS ADDED TO THE (GLOBAL) COMMAND
BUFFER. IF THERE
ISN'T ENOUGH ROOM, THE BUFFER IS FLUSHED (VIA FDEVOTP)
FIRST.

ARGUMENTS:

D = PD *
S = CHAR *
L = INT

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
TIME - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS
VTICOM - VTI COMMUNICATION DEFINITIONS
FUNCTS - FUNCTION DEFINITIONS
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

MEMCPY
PDVOTP - PUT DEVICE OUTPUT

CALLED DIRECTLY BY:

OISCR/FVTBUF - FILL VTI BUFFER
OISCR/PROCFLD - PROCESS FIELD
OISCR/PROCWIN - PROCESS WINDOW

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: OISCR/CMPALL
PURPOSE: COMPUTE ALL CALCULATED FIELDS
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: OISCR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

----- SYNOPSIS

STATIC CHAR *CMPALL(DP)
FIELD *DP;

INPUTS:

DP - POINTER TO FIRST FIELD TO BE COMPUTED

OUTPUTS:

RETURNS A RETURN CODE

DESCRIPTION

ALL FIELDS SUBORDINATE TO THE GIVEN FIRST FIELD ARE
EXAMINED; IF A FIELD
HAS BEEN CHANGED (EITHER ON INPUT OR OUTPUT), ALL OF THE
FIELD WHICH ARE
DEPENDENT ON IT ARE RECALCULATED IF THEY HAVE NOT ALREADY
BEEN CALCULATED
(A FIELD IS CONSIDERED TO HAVE ALREADY BEEN CALCULATED IF
IT HAS BEEN
CHANGED ON OUTPUT).

ARGUMENTS:

DP = FIELD *

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
STD IO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
TIME - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS
VTICOM - VTI COMMUNICATION DEFINITIONS
FUNCTS - FUNCTION DEFINITIONS
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

OISCR/CMPALL - COMPUTE ALL CALCULATED FIELDS
CMPFLD - COMPUTE FIELD

CALLED DIRECTLY BY:

OISCR/DSPSCR - DISPLAY SCREEN
OISCR/CMPALL - COMPUTE ALL CALCULATED FIELDS

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: OISCR/CNGMSG
PURPOSE: CHECK FOR AND PROCESS CHANGE MESSAGE REQUESTS
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: OISCR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

STATIC CHAR *CNGMSG(COUNT)

INT COUNT;

INPUTS:

COUNT - MESSAGE COUNT RECEIVED FROM DEVICE

OUTPUTS:

RETURNS A RETURN CODE

DESCRIPTION

CNGMSG ALLOWS THE USER TO SCROLL THROUGH THE MESSAGES IN THE MESSAGE LINE BUFFER. IT PUTS THE MESSAGE CORRESPONDING TO THE NUMBER SPECIFIED BY THE USER INTO THE MESSAGE LINE ON THE TERMINAL SCREEN.

ARGUMENTS:

COUNT = INT

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
TIME - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS
VTICOM - VTI COMMUNICATION DEFINITIONS
FUNCTS - FUNCTION DEFINITIONS
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

SYSMSG - SYSTEM MESSAGE ROUTINE

PS 620344200
30 September 1990

CALLED DIRECTLY BY:

OISCR/EVTBUF - EMPTY VTI BUFFER

FORM PROCESSOR Module Documentation

NAME: OISCR/DSPSCR
PURPOSE: DISPLAY SCREEN
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: OISCR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS
STATIC VOID DSPSCR()

DESCRIPTION
DISPLAY ALL INTERNALLY CALCULATED FIELDS INCLUDING ALL
USER CALCULATED
FIELDS.

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
TIME - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS
VTICOM - VTI COMMUNICATION DEFINITIONS
FUNCTS - FUNCTION DEFINITIONS
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

OISCR/CMPALL - COMPUTE ALL CALCULATED FIELDS
PDATA - PUT FORM DATA
SPRINTF
LOCALTIME
TIME

CALLED DIRECTLY BY:

OISCR/FVTBUF - FILL VTI BUFFER

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: OISCR/EVTBUF
PURPOSE: EMPTY VTI BUFFER
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: OISCF
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
STATIC CHAR *EVTBUF(PDPTR, VBPTR, VBEND)
    PD *PDPTR;
    CHAR *VBPTR, *VBEND;
```

INPUTS:

PDPTR - POINTER TO PHYSICAL DEVICE INPUT IS FROM
VBPTR - POINTER TO START OF INPUT BUFFER
VBEND - POINTER TO (CHARACTER PAST) END OF INPUT BUFFER

OUTPUTS:

RETURNS A RETURN CODE

DESCRIPTION

GETS DATA FROM THE VIRTUAL TERMINAL BUFFER AND STORES IT
IN THE FPD

DATA STRUCTURE.

ARGUMENTS:

PDPTR =	PD *
VBPTR =	CHAR *
VBEND =	CHAR *

INCLUDE FILES:

STDYTP	- STANDARD TYPE DEFINITIONS
STDIO	- **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE	- **** PURPOSE NOT FOUND BY STRIPPER ****
TIME	- **** PURPOSE NOT FOUND BY STRIPPER ****
FPD	- FORM PROCESSOR DATA
FPCODE	- FORM PROCESSOR RETURN CODES
FPPARM	- FORM PROCESSOR PARAMETERS
VTICOM	- VTI COMMUNICATION DEFINITIONS
FUNCTS	- FUNCTION DEFINITIONS
CTLCHR	- CONTROL CHARACTERS

ROUTINES CALLED:

STRASN
MEMSET
CBPTR
MEMCPY
BLEN
ISPRINT
FNFPWN - FIND FORM PROCESSOR WINDOW
MATOI
OISCR/CNGMSG - CHECK FOR AND PROCESS CHANGE MESSAGE REQUESTS
SYMSG - SYSTEM MESSAGE ROUTINE
GVTICMD
BLDCMD

CALLED DIRECTLY BY:

INSCR - INPUT SCREEN

FORM PROCESSOR Module Documentation

NAME: OISCR/FVTBUF
PURPOSE: FILL VTI BUFFER
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: OISCR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

STATIC CHAR *FVTBUF(INPFLG)
BOOL INPFLG;

INPUTS:

INPFLG - INPUT FLAG (TRUE IF INPUT IS TO BE ENABLED)

OUTPUTS:

RETURNS A RETURN CODE

DESCRIPTION

BUILDS A VTI DATA STREAM FROM THE DISPLAY LIST.

ARGUMENTS:

INPFLG = BOOL

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
TIME - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS
VTICOM - VTI COMMUNICATION DEFINITIONS
FUNCTS - FUNCTION DEFINITIONS
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

DSPMSG
PDVOTP - PUT DEVICE OUTPUT
OISCR/PROCWIN - PROCESS WINDOW
OISCR/PROCFLD - PROCESS FIELD
STRLEN

OISCR/ADDCMD - ADD COMMAND TO BUFFER
SPRINTF
OISCR/DSPSCR - DISPLAY SCREEN

CALLED DIRECTLY BY:

OUTSCR - OUTPUT SCREEN
ONWISC - OUTPUT (NO WAIT) / INPUT SCREEN
INSCR - INPUT SCREEN

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: OISCR/GATINF
PURPOSE: GET ATTRIBUTE INFO
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: OISCR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

----- SYNOPSIS

```
VOID GATINF(DP, ATP)
    FIELD *DP;
    ATTDEF *ATP;
```

INPUTS:

DP - FIELD TO GET ATTRIBUTE INFORMATION FOR

OUTPUTS:

ATP - POINTER TO ATTRIBUTES

DESCRIPTION

RETURNS THE CURRENT ATTRIBUTES OF THE SPECIFIED FIELD,
TAKING TEMPORARY
ATTRIBUTES INTO ACCOUNT.

ARGUMENTS:

DP = FIELD *
ATP = ATTDEF *

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
STD IO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
TIME - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS
VTICOM - VTI COMMUNICATION DEFINITIONS
FUNCTS - FUNCTION DEFINITIONS
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

MABSAT - MAP ABSOLUTE ATTRIBUTE
STRASN

CALLED DIRECTLY BY:

OISCR/PROCFLD - PROCESS FIELD

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: OISCR/PROCFLD
PURPOSE: PROCESS FIELD
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: OISCR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

----- SYNOPSIS

```
STATIC VOID PROCFLD(DP, CNGALL, DROW, DCOL)
    FIELD *DP;
    BOOL CNGALL;
    INT DROW, DCOL;
```

INPUTS:

```
DP      - FIELD TO PROCESS
CNGALL  - GLOBAL CHANGE FLAG
DROW    - OFFSET TO BE ADDED TO FIELD ROW
DCOL    - OFFSET TO BE ADDED TO FIELD COLUMN
```

DESCRIPTION

DO DEFINE FIELD OR DEFINE WINDOW FOR CURRENT FIELD AND
COMPUTE DEFAULT
CURSOR POSITION.

ARGUMENTS:

DP = FIELD *
CNGALL = BOOL
DROW = INT
DCOL = INT

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
TIME - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS
VTICOM - VTI COMMUNICATION DEFINITIONS
FUNCTS - FUNCTION DEFINITIONS
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

POSCUR - POSITION CURSOR
CBPTR
BLEN
STRCAT
OISCR/PROCFLD - PROCESS FIELD
ABS
MIN
OISCR/ADDCMD - ADD COMMAND TO BUFFER
SYMSG - SYSTEM MESSAGE ROUTINE
PDVOTP - PUT DEVICE OUTPUT
STRLEN
SPRINTF
OISCR/GATINF - GET ATTRIBUTE INFO

CALLED DIRECTLY BY:

OISCR/FVTBUF - FILL VTI BUFFER
OISCR/PROCFLD - PROCESS FIELD
OISCR/PROCWIN - PROCESS WINDOW

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: OISCR/PROWIN
PURPOSE: PROCESS WINDOW
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: OISCR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
STATIC VOID PROCWIN(DP, CNGALL, DROW, DCOL)
    FIELD *DP;
    BOOL CNGALL;
    INT DROW, DCOL;
```

INPUTS:

```
DP      - FIELD TO PROCESS
CNGALL  - GLOBAL CHANGE FLAG
DROW    - OFFSET TO BE ADDED TO FIELD ROW
DCOL    - OFFSET TO BE ADDED TO FIELD COLUMN
```

DESCRIPTION

DO SET WINDOW COMMAND FOR CURRENT WINDOW AND PROCESS
CONTAINED FIELDS.

ARGUMENTS:

```
DP =          FIELD *
CNGALL =       BOOL
DROW =         INT
DCOL =         INT
```

INCLUDE FILES:

```
STDTyp      - STANDARD TYPE DEFINITIONS
STDIO       - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE       - **** PURPOSE NOT FOUND BY STRIPPER ****
TIME        - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD         - FORM PROCESSOR DATA
FPCODE      - FORM PROCESSOR RETURN CODES
FPPARM      - FORM PROCESSOR PARAMETERS
VTICOM      - VTI COMMUNICATION DEFINITIONS
FUNCTS      - FUNCTION DEFINITIONS
CTLCHR      - CONTROL CHARACTERS
```

ROUTINES CALLED:

OISCR/PROCFLD - PROCESS FIELD
STRLEN
OISCR/ADDCMD - ADD COMMAND TO BUFFER
SPRINTF
OISCR/PROCWIN - PROCESS WINDOW
ABS

CALLED DIRECTLY BY:

OISCR/FVTBUF - FILL VTI BUFFER
OISCR/PROCWIN - PROCESS WINDOW

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: OISCR/RSTINP
PURPOSE: RESET INPUT FLAGS
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: OISCR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

STATIC VOID RSTINP(DP)
FIELD *DP;

INPUTS:

DP - POINTER TO FIELD TO RESET INPUT FLAGS FOR

OUTPUTS:

NONE

DESCRIPTION

RSTINP RESETS THE CHANGED ON INPUT FLAGS FOR A FIELD AND
ITS CHILDREN

ARGUMENTS:

DP = FIELD *

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
TIME - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS
VTICOM - VTI COMMUNICATION DEFINITIONS
FUNCTS - FUNCTION DEFINITIONS
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

OISCR/RSTINP - RESET INPUT FLAGS
BLEN

PBPTR
CBPTR
MEMCPY

CALLED DIRECTLY BY:

OISCR/RSTINP - RESET INPUT FLAGS
ONWISC - OUTPUT (NO WAIT) / INPUT SCREEN

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: OISCR/RSTMAT
PURPOSE: RESET TEMPORAY ATTRIBUTES
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: OISCR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

----- SYNOPSIS

STATIC VOID RSTMAT(DP)
FIELD *DP;

INPUTS:

DP - POINTER TO FIELD TO RESET TEMPORARY ATTRIBUTES FOR

DESCRIPTION

RSTMAT RESETS THE TEMPORARY ATTRIBUTE FOR A FIELD AND ITS
CHILDREN

ARGUMENTS:

DP = FIELD *

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
TIME - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS
VTICOM - VTI COMMUNICATION DEFINITIONS
FUNCTS - FUNCTION DEFINITIONS
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

OISCR/RSTMAT - RESET TEMPORAY ATTRIBUTES

CALLED DIRECTLY BY:

OISCR/RSTMAT - RESET TEMPORAY ATTRIBUTES
INSCR - INPUT SCREEN

FORM PROCESSOR Module Documentation

NAME: OISCR/SETWIN
PURPOSE: SET WINDOW
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: OISCR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

----- SYNOPSIS

```
STATIC VOID SETWIN(DP, WP, ATT)
    FIELD *DP, *WP;
    ATTMAP *ATT;
```

INPUTS

DP - POINTER TO TOP FIELD TO CHANGE ATTRIBUTE OF
WP - POINTER TO LAST FIELD TO CHANGE ATTRIBUTE OF
ATT - ATTRIBUTE TO SET

DESCRIPTION

STARTING WITH THE SPECIFIED FIELD, SET IT AND ALL OF ITS
DECENDANTS'
TEMPORARY ATTRIBUTES, STOPPING AT THE SPECIFIED LAST
FIELD.

ARGUMENTS:

DP = FIELD *
WP = FIELD *
ATT = ATTMAP *

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
TIME - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS
VTICOM - VTI COMMUNICATION DEFINITIONS
FUNCTS - FUNCTION DEFINITIONS
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

OISCR/SETWIN - SET WINDOW

CALLED DIRECTLY BY:

OISCR/SETWIN - SET WINDOW
ONWISC - OUTPUT (NO WAIT) / INPUT SCREEN

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: ONWISC
PURPOSE: OUTPUT (NO WAIT) / INPUT SCREEN
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: BOOL ()
SOURCE FILE: OISCR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
BOOL ONWISC(PATH, CODEP)
    EPATH PATH;
    CHAR CODEP[];
```

INPUTS:

PATH - WINDOW TO ACCEPT INPUT FROM

OUTPUTS:

CODEP - RETURN CODE
RETURNS ALSO SUCCESS = 0 / FAILURE = 1

DESCRIPTION

DISPLAYS THE FORMS ON THE DISPLAY LIST AND SETS INPUT
PENDING ON THE
SPECIFIED WINDOW.

ARGUMENTS:

PATH = EPATH
CODEP = CHAR []

INCLUDE FILES:

```
STD TYP - STANDARD TYPE DEFINITIONS
STD IO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
TIME - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS
VTICOM - VTI COMMUNICATION DEFINITIONS
FUNCTS - FUNCTION DEFINITIONS
CTLCHR - CONTROL CHARACTERS
```

ROUTINES CALLED:

ESCPY - EXTERNAL STRING COPY
PTHPTR - GET PATH POINTER
OISCR/SETWIN - SET WINDOW
OISCR/RSTINP - RESET INPUT FLAGS
OISCR/FVTBUF - FILL VTI BUFFER
MEMCPY
SYSMSG - SYSTEM MESSAGE ROUTINE
GATDEF - GET ATTRIBUTE DEFINITION

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES
MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS
UIS - USR INTERFACE SERVICES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: OPNFRM
PURPOSE: OPEN FORM
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FORTRAN VOID ()
SOURCE FILE: OPNFRM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
FORTRAN VOID OPNFRM(FRMNAM, CODEP)
ENAME FRMNAM;
CHAR CODEP[];
```

INPUTS:

FRMNAM - NAME OF FORM TO OPEN

OUTPUTS:

CODEP - RETURN CODE

DESCRIPTION

OPNFRM IS USED TO RETRIEVE A FORM FROM A LIBRARY. NORMAL
SEARCH
RULES ARE USED TO FIND THE FORM. THE FORM IS MADE ACTIVE
AND THE
DEFAULT FIELD DATA IS MADE AVAILABLE. IF THE FORM
CONTAINS SUBFORMS
THEY ARE ALSO OPENED. THE FORM IS NOT DISPLAYED AT THIS
POINT.

ARGUMENTS:

FRMNAM =	ENAME
CODEP =	CHAR []

INCLUDE FILES:

STDTyp	- STANDARD TYPE DEFINITIONS
STDIO	- **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE	- **** PURPOSE NOT FOUND BY STRIPPER ****
FPD	- FORM PROCESSOR DATA
FFFV2	- FORM FILE FORMAT - VERSION 2
FPCODE	- FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

ESCPY - EXTERNAL STRING COPY
STRUPC
GOFPTR - GET OPEN FROM POINTER
SYSMSG - SYSTEM MESSAGE ROUTINE
OPNFRM/PFREC - PROCESS FORM RECORD
DELFLD - DELETE FIELD
FCLOSE
MEMCPY
SPRINTF
FOPEN

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES
COPFRM - COPY FORM
OPNFRM/PFR - PROCESS FORM
UIS - USR INTERFACE SERVICES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: OPNFRM/BDBUFF
PURPOSE: BUILD DEFAULT BUFFER
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: OPNFRM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

----- DESCRIPTION

BDBUFF GETS DEFAULT FIELD VALUES FROM THE FORM DESCRIPTION
FILE AND USES THEM TO BUILD THE DEFAULT BUFFER.

ARGUMENTS:

FILE = FILE *
FRP = FRMREC *
FP = FIELD *

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FFFV2 - FORM FILE FORMAT - VERSION 2
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

FREE
OPNFRM/BFLDDB - BUILD FIELD DEFAULT BUFFER
GETC
ISCNTRL
FEOF
FERROR
SYSMSG - SYSTEM MESSAGE ROUTINE
MALLOC

CALLED DIRECTLY BY:

OPNFRM/PFREC - PROCESS FORM RECORD

PS 620344200
30 September 1990

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: OPNFRM/BFLDDB
PURPOSE: BUILD FIELD DEFAULT BUFFER
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: OPNFRM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

DESCRIPTION

BFLDDB BUILDS THE DEFAULT BUFFER FOR A FIELD. IT HAS TO
TO PROPAGATE THE VALUE IN THE DEFAULT BUFFER FOR EACH ITEM
OF AN ARRAY.

ARGUMENTS:

DP = FIELD *
TYPE = CHAR
TPP = CHAR **

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FFFV2 - FORM FILE FORMAT - VERSION 2
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

OPNFRM/BFLDDB - BUILD FIELD DEFAULT BUFFER
MEMCPY
BLEN
SYSMSG - SYSTEM MESSAGE ROUTINE
MALLOC

CALLED DIRECTLY BY:

OPNFRM/BDBUFF - BUILD DEFAULT BUFFER
OPNFRM/BFLDDB - BUILD FIELD DEFAULT BUFFER

PS 620344200
30 September 1990

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: OPNFRM/BRPNOD
PURPOSE: BUILD RELATIVE POSITION NODE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: OPNFRM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

DESCRIPTION

ARGUMENTS:

FILE = FILE *
DP = FIELD *

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FFFV2 - FORM FILE FORMAT - VERSION 2
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

FERROR
ABORT
FNDFLD - FIND FIELD
STRASN
SYSMSG - SYSTEM MESSAGE ROUTINE
MALLOC
FREAD

CALLED DIRECTLY BY:

OPNFRM/PFREC - PROCESS FORM RECORD

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: OPNFRM/BTBUFF
PURPOSE: BUILD TEXT BUFFER
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: OPNFRM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

DESCRIPTION

BTBUFF GETS TEXT INFORMATION FROM THE FORM DESCRIPTION
FILE
AND USES IT TO BUILD THE TEXT BUFFER.

ARGUMENTS:

FILE = FILE *
FRP = FRMREC *
FP = FIELD *

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FFFV2 - FORM FILE FORMAT - VERSION 2
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

FERROR
GETC
ISCNTRL
SYSMSG - SYSTEM MESSAGE ROUTINE
MALLOC

CALLED DIRECTLY BY:

OPNFRM/PFREC - PROCESS FORM RECORD

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: OPNFRM/PARY
PURPOSE: PROCESS ARRAY
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: OPNFRM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

----- DESCRIPTION

PARY LOADS THE ARRAY INFORMATION FOR A FIELD FROM THE FORM
DESCRIPTION FILE INTO THE ARYTYP DATA STRUCTURE IN FD.H.

ARGUMENTS:

DRP = FLDREC *
NDP = FIELD **
PP = FIELD *
FLDNAM = NAME
ROW = INT
COL = INT
LEVEL = INT

INCLUDE FILES:

STDTPY - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FFFV2 - FORM FILE FORMAT - VERSION 2
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

OPNFRM/PARY - PROCESS ARRAY
ABS
MAX
MAKFLD - MAKE FIELD
GATDEF - GET ATTRIBUTE DEFINITION
SYSMSG - SYSTEM MESSAGE ROUTINE
OPNFRM/PWIN - PROCESS WINDOW
OPNFRM/PFRM - PROCESS FORM
OPNFRM/PITM - PROCESS ITEM

CALLED DIRECTLY BY:

OPNFRM/PDREC - PROCESS FIELD RECORD
OPNFRM/PAR - PROCESS ARRAY

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: OPNFRM/PDREC
PURPOSE: PROCESS FIELD RECORD
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: OPNFRM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

----- DESCRIPTION

PDREC LOADS THE FIELD RECORDS FROM THE FORM DESCRIPTION
INTO THE FIELD DATA STRUCTURE.

ARGUMENTS:

FILE = FILE *
FRP = FRMREC *
PP = FIELD *

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FFV2 - FORM FILE FORMAT - VERSION 2
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

OPNFRM/PARY - PROCESS ARRAY
ESCPY - EXTERNAL STRING COPY
SYSMSG - SYSTEM MESSAGE ROUTINE
FREAD

CALLED DIRECTLY BY:

OPNFRM/PFREC - PROCESS FORM RECORD

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: OPNFRM/PFREC
PURPOSE: PROCESS FORM RECORD
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: OPNFRM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

----- DESCRIPTION

PFREC LOADS THE FORM RECORDS FROM THE FORM DESCRIPTION
FILE INTO THE FORM DATA STRUCTURE.

ARGUMENTS:

FILE = FILE *
DP = FIELD **
FNAME = NAME

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FFFV2 - FORM FILE FORMAT - VERSION 2
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

OPNFRM/BRPNOD - BUILD RELATIVE POSITION NODE
OPNFRM/BDBUFF - BUILD DEFAULT BUFFER
OPNFRM/BTBUFF - BUILD TEXT BUFFER
OPNFRM/PDREC - PROCESS FIELD RECORD
OPNFRM/PTREC - PROCESS TEXT RECORD
MAKFLD - MAKE FIELD
GATDEF - GET ATTRIBUTE DEFINITION
STRCMP
ESCPY - INTERNAL STRING COPY
SYSMSG - SYSTEM MESSAGE ROUTINE
FREAD

CALLED DIRECTLY BY:

OPNFRM - OPEN FORM

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: OPNFRM/PFRM
PURPOSE: PROCESS FORM
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: OPNFRM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

----- DESCRIPTION

THIS ROUTINE IS USED IN THE SITUATION WHERE WE ENCOUNTER
A FORM WITHIN A FORM. THE NEW FORM IS OPENED RECURSIVELY
AND THEN COPIED TO THE APPROPRIATE PLACE IN THE CONTAINING
FORM.

ARGUMENTS:

DRP = FLDREC *
NDP = FIELD **
PP = FIELD *
FLDNAM = NAME
ROW = INT
COL = INT

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
STD IO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FFFV2 - FORM FILE FORMAT - VERSION 2
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

COPFLD - COPY FIELD
MAKFLD - MAKE FIELD
SYSMSG - SYSTEM MESSAGE ROUTINE
OPNFRM - OPEN FORM
GOFPTR - GET OPEN FROM POINTER

CALLED DIRECTLY BY:

OPNFRM/PAR - PROCESS ARRAY

PS 620344200
30 September 1990

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: OPNFRM/PITM
PURPOSE: PROCESS ITEM
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: OPNFRM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

----- DESCRIPTION

PITM PROCESSES ITEM INFORMATION FROM THE FORM DESCRIPTION
FILE PUTS IT INTO THE ITEM STRUCTURE IN FD.H.

ARGUMENTS:

DRP = FLDREC *
NDP = FIELD **
PP = FIELD *
FLDNAM = NAME
ROW = INT
COL = INT

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FFFV2 - FORM FILE FORMAT - VERSION 2
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

MEMCMP
MEMCPY
SYSMSG - SYSTEM MESSAGE ROUTINE
MALLOC
STRLEN
MAKFLD - MAKE FIELD
GATDEF - GET ATTRIBUTE DEFINITION
ESCPY - EXTERNAL STRING COPY

CALLED DIRECTLY BY:

OPNFRM/PAR - PROCESS ARRAY

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: OPNFRM/PTREC
PURPOSE: PROCESS TEXT RECORD
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: OPNFRM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

DESCRIPTION

PTREC LOADS THE TEXT RECORDS FROM THE FORM DESCRIPTION
FILE
INTO THE TEXT DATA STRUCTURE.

ARGUMENTS:

FILE = FILE *
FRP = FRMREC *
FP = FIELD *

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FFV2 - FORM FILE FORMAT - VERSION 2
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

MALLOC
SYSMSG - SYSTEM MESSAGE ROUTINE
FREAD

CALLED DIRECTLY BY:

OPNFRM/PFREC - PROCESS FORM RECORD

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: OPNFRM/PWIN
PURPOSE: PROCESS WINDOW
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: OPNFRM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

----- DESCRIPTION

PWIN GETS WINDOW INFORMATION FROM THE FORM DESCRIPTION FILE
AND INSERTS IT INTO THE WINTYP DATA STRUCTURE IN FD.H.

ARGUMENTS:

DRP = FLDREC *
NDP = FIELD **
PP = FIELD *
FLDNAM = NAME
ROW = INT
COL = INT

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
STD IO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYP E - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FFFV2 - FORM FILE FORMAT - VERSION 2
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

STRASN
MAKFLD - MAKE FIELD
GATDEF - GET ATTRIBUTE DEFINITION
ESCPY - EXTERNAL STRING COPY

CALLED DIRECTLY BY:

OPNFRM/PAR - PROCESS ARRAY

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: OPNLDV
PURPOSE: OPEN LOGICAL DEVICE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FORTRAN VOID ()
SOURCE FILE: OPNLDV
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
FORTRAN VOID OPNLDV(LDWNID,RCODE)
      INT      *LDWNID;
      CHAR     RCODE[];
```

INPUTS/OUTPUTS:

INPUTS:

ADDRESS OF: LOGICAL DEVICES TOP WINDOW ID
RETURN CODE

OUTPUTS:

LDWNID - LOGICAL DEVICES TOP WINDOW ID
RCODE - RETURN CODE

DESCRIPTION

THIS MODULE OPENS A LOGICAL DEVICE. IF IT FAILS IT
RETURNS NFPDSTRC ERROR.

ARGUMENTS:

LDWNID = INT *
RCODE = CHAR []

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

STUPFP - SET UP FORM PROCESSOR DATA STRUCTURES
MAKFPD - MAKE FORM PROCESSOR DATA (LOGICAL DEVICE
 STRUCTURE)
MEMCPY

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: OUTSCR
PURPOSE: OUTPUT SCREEN
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FORTRAN VOID ()
SOURCE FILE: OISCR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

----- SYNOPSIS

```
FORTRAN VOID OUTSCR(PATH, CODEP)
    EPATH PATH;
    CHAR CODEP[];
```

INPUTS:

PATH - INPUT WINDOW PATH NAME - NOT USED

OUTPUTS:

CODEP - RETURN CODE

DESCRIPTION

DISPLAY ALL FORMS ON THE DISPLAY LIST.

ARGUMENTS:

PATH = EPATH
CODEP = CHAR []

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
TIME - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS
VTICOM - VTI COMMUNICATION DEFINITIONS
FUNCTS - FUNCTION DEFINITIONS
CTLCHR - CONTROL CHARACTERS

ROUTINES CALLED:

MEMCPY
OISCR/FVTBUF - FILL VTI BUFFER

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES
MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS
UIS - USR INTERFACE SERVICES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: PARFQN
PURPOSE: PARSE FULLY QUALIFIED NAME
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: PARFQN
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

FORTRAN VOID PARFQN(FQN, LSTFTP, LEVEL, PARNAM, PARTYP, RCODE)

EPATH FQN;
CHAR *LSTFTP;
SHORT *LEVEL;
EPATH PARNAM;
CHAR *PARTYP;
CHAR *RCODE;

INPUTS:

FQN = FULLY QUALIFIED NAME TO BE PARSED.
LSTFTP = TYPE OF THE LAST FIELD IN FULLY QUALIFIED NAME
LEVEL = LEVEL OF FIELD INTERESTED IN OBTAINING
0 = LAST LEVEL OF FULLY QUALIFIED NAME
-1 = FIRST FROM LAST LEVEL OF FULLY QUALIFIED NAME
-2 = SECOND FROM LAST LEVEL OF FULLY QUALIFIED NAME
ETC....
1 = FIRST LEVEL OF FULLY QUALIFIED NAME
2 = SECOND LEVEL OF FULLY QUALIFIED NAME
ETC....

OUTPUTS:

PARNAM = PARSED NAME
PARTYP = TYPE OF THE FIELD WITH PARSED NAME
RCODE = RETURN CODE

DESCRIPTION

PARFQN WILL RETURN THE NAME OF THE FIELD AND ITS TYPE AT A SPECIFIED LEVEL OF A SPECIFIED FULLY QUALIFIED NAME GIVEN THE TYPE OF THE LAST FIELD OF THIS FULLY QUALIFIED NAME.

ARGUMENTS:

FQN =
LSTFTP = EPATH
CHAR *

LEVEL = SHORT *
PARNAM = EPATH
PARTYP = CHAR *
RCODE = CHAR *

INCLUDE FILES:

STDTPY - STANDARD TYPE DEFINITIONS
FPCODE - FORM PROCESSOR RETURN CODES
FPD - FORM PROCESSOR DATA

ROUTINES CALLED:

ESCPY - EXTERNAL STRING COPY
STRCHR
MEMCPY
STRLEN
MEMSET
SYSMSG - SYSTEM MESSAGE ROUTINE
STRRCHR

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: PDATA
PURPOSE: PUT FORM DATA
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FORTRAN VOID ()
SOURCE FILE: PDATA
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
FORTRAN VOID PDATA(EWPATH, FDATA, CODEP)
    EPATH EWPATH;
    CHAR *FDATA;
    CHAR CODEP[];
```

INPUTS:

EWPATH - PATH NAME
FDATA - PATH DATA

OUTPUTS:

CODEP - RETURN CODE

DESCRIPTION

USE PDATA TO PUT DATA ON A FORM, FIELD, ARRAY OR WINDOW.

ARGUMENTS:

EWPATH = EPATH
FDATA = CHAR *
CODEP = CHAR []

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

ESCPY - EXTERNAL STRING COPY
PTHPTR - GET PATH POINTER
PDATA/PUTBUF - PUT BUFFER
MEMCPY

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES
OISCR/DSPSCR - DISPLAY SCREEN
UIS - USR INTERFACE SERVICES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: PDATA/PUTBUF
PURPOSE: PUT BUFFER
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: PDATA
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

DESCRIPTION

PUTBUF PUTS DATA IN THE BUFFER AND UPDATES THE BUFFER
POINTER.

ARGUMENTS:

DP = FIELD *

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

PDATA/PUTBUF - PUT BUFFER
SYSMSG - SYSTEM MESSAGE ROUTINE
ISPRINT
MEMCPY
MEMCMP
BLEN
CBPTR

CALLED DIRECTLY BY:

PDATA/PUTBUF - PUT BUFFER
PDATA - PUT FORM DATA

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: PDVOTP
PURPOSE: PUT DEVICE OUTPUT
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: PDVOTP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
VOID PDVOTP(PDPTR, BUFF, LEN)
    PD *PDPTR;
    CHAR *BUFF;
    INT LEN;
```

INPUTS/OUTPUTS:

INPUTS:

```
PD      *PDPTR = POINTER TO DEVICE SENDING BUFF TO
CHAR    *BUFF  = BUFFER SENDING TO VIRTUAL TERMINAL
INT     LEN    = LENGTH OF THIS BUFFER
```

OUTPUTS:

NONE

DESCRIPTION

PDVOTP SENDS MESSAGES TO LOW LEVEL DRIVERS.

ARGUMENTS:

```
PDPTR = PD *
BUFF = CHAR *
LEN = INT
```

INCLUDE FILES:

```
STDTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
VTICOM - VTI COMMUNICATION DEFINITIONS
NTM - NTM INTERFACE INCLUDE FILE
FPCODE - FORM PROCESSOR RETURN CODES
```

ROUTINES CALLED:

MEMSET
MEMCPY
STRLEN
FWRITE
FCLOSE
SYSMSG - SYSTEM MESSAGE ROUTINE
NSEND
MEMCMP

CALLED DIRECTLY BY:

GDVINP - GET DEVICE INPUT
MAKUSR - MAKE USER
OISCR/FVTBUF - FILL VTI BUFFER
OISCR/PROCFLD - PROCESS FIELD
OISCR/ADDCMD - ADD COMMAND TO BUFFER
RMVFPD - REMOVE FORM PROCESSOR DATA STRUCTURE
UIS/PRCINP - PRCESS INPUT
UIS - USR INTERFACE SERVICES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: PMSGC
PURPOSE: PUT MESSAGE LINE CODE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FORTRAN VOID ()
SOURCE FILE: PMSGC
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

FORTRAN VOID PMSGC(CODEP)
CHAR CODEP[];

INPUTS:

CODEP - RETURN CODE TO DISPLAY MESSAGE FOR

DESCRIPTION

PMSGC MATCHES A PREDETERMINED MESSAGE WITH RCODE. IT THEN
INSERTS THE MESSAGE INTO THE MESSAGE-LINE PORTION OF THE
BUFFER SO THAT THE MESSAGE WILL BE DISPLAYED THE NEXT TIME
A SCREEN IS SENT TO THE TERMINAL. FOLLOW THIS CALL WITH A
CALL TO OUTSCR OR OISCR FOR IMMEDIATE TRANSMISSION.

ARGUMENTS:

CODEP = CHAR []

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA

ROUTINES CALLED:

FNDMSG - FIND MESSAGE
PMSGC - PUT MESSAGE LINE STRING

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES
MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS
INSCR - INPUT SCREEN

SYSMSG - SYSTEM MESSAGE ROUTINE
UIS/STRTP - START APPLICATION
UIS - USR INTERFACE SERVICES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: PMSGLS
PURPOSE: PUT MESSAGE LINE STRING
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FORTRAN VOID ()
SOURCE FILE: PMSGLS
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
FORTRAN VOID PMSGLS(MSGSTR)
    EMSG MSGSTR;
```

INPUTS:

MSGSTR - MESSAGE TO DISPLAY

DESCRIPTION

PMSGLS INSERTS MSG-STRING INTO THE MESSAGE-LINE PORTION OF THE BUFFER. THE MESSAGE-LINE WILL BE DISPLAYED THE NEXT TIME A SCREEN IS SENT TO THE TERMINAL. FOLLOW THIS CALL WITH A CALL TO OUTSCR OR OISCR FOR IMMEDIATE TRANSMISSION.

ARGUMENTS:

MSGSTR = EMSG

INCLUDE FILES:

```
STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
```

ROUTINES CALLED:

```
MALLOC
FUISWN - FIND UIS WINDOW
ESCPY - EXTERNAL STRING COPY
STRLEN
MEMSET
ISPRINT
```

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES
PMSGLC - PUT MESSAGE LINE CODE
RMVAP - REMOVE APPLICATION
UIS/STRTP - START APPLICATION
UIS/STRTPD - START PHYSICAL DEVICE
UIS/PRCWND - PRCESS WINDOW
UIS - USR INTERFACE SERVICES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: POSCUR
PURPOSE: POSITION CURSOR
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: POSCUR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
VOID POSCUR(DP,RELPOS,ABSPOS)
    REGISTER FIELD *DP;
    POSITION        *RELPOS;
    POSITION        *ABSPOS;
```

INPUTS/OUTPUTS:

INPUTS:

DP - FIELD WHOSE ROW AND COL WANT TO ABSOLUTIZED
STRUCTURE CONTAINING:
RELATIVE ROW OF FIELD
RELATIVE COL OF FIELD

OUTPUTS:

POSITION OF CURSOR

DESCRIPTION

THIS MODUL CALUCULATES ABSOLUTE CURSOR FOR INSERTING INTO
OUTCUR, GOING
UP CHILD PARENT TREE AND ADDING EACH SUCCESSIVE PARENT'S
ROW AND COL
TO SUM OF CHILDS', CLIPPING POSITION TO BOUNDS OF PARENT
WINDOW.

ARGUMENTS:

DP = FIELD *
RELPOS = POSITION *
ABSPOS = POSITION *

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA

ROUTINES CALLED:

POSCUR/FNFITM - FIND FIRST ITEM OF FIELD
MAX
MIN

CALLED DIRECTLY BY:

OISCR/PROCFLD - PROCESS FIELD
PUTCUR - PUT CURSOR
PUTLOC - PUT LOCATION
RMVFPD - REMOVE FORM PROCESSOR DATA STRUCTURE

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: POSCUR/FNFITM
PURPOSE: FIND FIRST ITEM OF FIELD
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FIELD * ()
SOURCE FILE: POSCUR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

STATIC FIELD *FNFITM(DP)
REGISTER FIELD *DP;

INPUTS/OUTPUTS:

INPUTS:

DP - FIELD WHOSE A POINTER TO WHOSE FIRST ITEM IS DESIRED

OUTPUTS:

POINTER TO FIRST INPUT ITEM OF FIELDS CHILDREN IF ANY
OTHERWISE A NULL

DESCRIPTION

THIS MODUL FINDS THE THE FIEST ITEM OF A FIELD IF ANY

ARGUMENTS:

DP = FIELD *

INCLUDE FILES:

STD^{TYP} - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA

ROUTINES CALLED:

POSCUR/FNFITM - FIND FIRST ITEM OF FIELD

CALLED DIRECTLY BY:

POSCUR - POSITION CURSOR
POSCUR/FNFITM - FIND FIRST ITEM OF FIELD

PS 620344200
30 September 1990

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: PRNAP
PURPOSE: PRINT APPLICATION
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: PRNFLD
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

----- DESCRIPTION

FOR DEBUGGING:

PRINTS INFO FROM AP STURCTURE USING FPD POINTER

ARGUMENTS:

FPD = FPD *

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

PRINTF

CALLED DIRECTLY BY:

PRNPD - PRINT PHYSICAL DEVICE
PRNUSR - PRINT USER
PRNUID - PRINT UID

USED IN MAIN PROGRAM(S):

PRNUID - PRINT UID
PRNUSR - PRINT USER

FORM PROCESSOR Module Documentation

NAME: PRNDSP
PURPOSE: PRINT DISPLAY LIST
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: PRNFLD
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

DESCRIPTION

FOR DEBUGGING:

PRINTS A DISPLAY LIST CALLS PRNFLD WITH DSPLST

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

PRNFLD - PRINT FIELD

FORM PROCESSOR Module Documentation

NAME: PRNFLD
PURPOSE: PRINT FIELD
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: PRNFLD
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

PRNFLD(DP, CHLDFLG)

FIELD *DP;

BOOL CHLDFLG;

INPUTS:

DP - POINTER TO FIRST FIELD TO PRINT

CHLDFLG - FLAG - WHETHER WANT TO LOOK AT CHILDREN OR
NOT

DESCRIPTION

FOR DEBUGGING:

PRINTS A FIELD AND ITS CONTENTS FOLLOWED BY ITS NEXT
FIELD, ETC.

ARGUMENTS:

DP = FIELD *
CHLDFLG = BOOL

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

PRNFLD - PRINT FIELD
DOWIND
DOITEM
DOATTR
PRINTF

CALLED DIRECTLY BY:

PRNFLD - PRINT FIELD
PRNDSP - PRINT DISPLAY LIST
PRNOPN - PRINT OPEN LIST

USED IN MAIN PROGRAM(S):

PRNDSP - PRINT DISPLAY LIST
PRNOPN - PRINT OPEN LIST

FORM PROCESSOR Module Documentation

NAME: PRNOPN
PURPOSE: PRINT OPEN LIST
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: PRNFLD
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

DESCRIPTION
FOR DEBUGGING:
PRINTS A OPEN LIST CALLS PRNFLD WITH OPNLST

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

PRNFLD - PRINT FIELD

FORM PROCESSOR Module Documentation

NAME: PRNPD
PURPOSE: PRINT PHYSICAL DE. ME
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: PRNFLD
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

DESCRIPTION
FOR DEBUGGING:
PRINTS INFO FROM PD STURCTURE USING PD POINTER

ARGUMENTS:

PD = PD *

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

PRNAP - PRINT APLICATION
PRINTF

CALLED DIRECTLY BY:

PRNUSR - PRINT USER
PRNUID - PRINT UID

USED IN MAIN PROGRAM(S):

PRNUID - PRINT UID
PRNUSR - PRINT USER

FORM PROCESSOR Module Documentation

NAME: PRNUID
PURPOSE: PRINT UID
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: PRNFLD
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

DESCRIPTION

FOR DEBUGGING:

PRINTS UID INFO & CALLS PRNAP AND PRNPD FOR ALL AP AN PD
ON UIS AP
AND PD LIST

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

PRINTF
PRNPD - PRINT PHYSICAL DEVICE
PRNAP - PRINT APPLICATION

FORM PROCESSOR Module Documentation

NAME: PRNUSR
PURPOSE: PRINT USER
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: PRNFLD
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

DESCRIPTION

FOR DEBUGGING:

PRINTS USER INFO & CALLS PRNAP AND PRNPD FOR ALL AP AN
PD FOR ALL USERS

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

PRNPD - PRINT PHYSICAL DEVICE
PRNAP - PRINT APPLICATION
PRINTF

FORM PROCESSOR Module Documentation

NAME: PTHPTR
PURPOSE: GET PATH POINTER
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: PTHPTR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
CHAR *PTHPTR(PATH, PTHPP, TOPFLD)
    PATH PATH;
    FIELD **PTHPP;
    FIELD *TOPFLD;
```

INPUTS:

PATH - PATH TO GET POINTER TO
TOPFLD - POINTER TO FIRST FIELD TO SEARCH FOR PATH

OUTPUTS:

PTHPP - ADDRESS OF POINTER TO SET

DESCRIPTION

PARSES A QUALIFIED NAME PASSED AS A CHARACTER STRING IN
PATH,
DETERMINES WHICH FIELD THIS NAME INDICATES AND RETURNS A
POINTER
TO THAT FIELD IN THE DISPLAY LIST AS A PARAMETER.
PTHPTR RETURNS A NULL IF SUCCESSFUL ELSE RETURNS A POINTER
TO AN ERROR CODE.

THE METHOD IS TO FORM A LIST OF ELEMENTS. ONE ELEMENT FOR
EACH
QUALIFIER IN THE QUALIFIED NAME. THE FORM HIERARCHY IS
SEARCHED
BY CHECKING IF A QUALIFIER MATCHES A FIELD NAME AT THE
CURRENT
LEVEL AND A LOWER LEVEL. DEPENDING ON THE FIELD TYPE,
PROCESSING
WILL BE DONE FOR AN ARRAY, ITEM, FORM OR WINDOW. FIELD
NAME
MATCHING AND FIELD TYPE PROCESSING REPEATS UNTIL THE END
OF THE
ELEMENT LIST IS REACHED. AT THIS POINT PROCEDURE FOUND
DETERMINES
IF A FIELD HAS ACTUALLY BEEN LOCATED.

ARGUMENTS:

PATH = PATH
PTHPP = FIELD **
TOPFLD = FIELD *

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
FPCODE - FORM PROCESSOR RETURN CODES
FPD - FORM PROCESSOR DATA
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

MALLOC
STRASN
ISDIGIT
SYSMSG - SYSTEM MESSAGE ROUTINE
PTHPTR/FIELD - MATCH FIELD
FREE
ISALNUM
TOUPPER

CALLED DIRECTLY BY:

ADDELM - ADD ELEMENT
ADDFRM - ADD FORM TO WINDOW
GDATA - GET DATA
GDATLN - GET DATA LENGTH
GETATT - GET ATTRIBUTE
GETBAK - GET BACKGROUND ATTRIBUTE
GPAGE - GET PAGE
GWINDO - GET WINDOW
ONWISC - OUTPUT (NO WAIT) / INPUT SCREEN
PDATA - PUT FORM DATA
PUTATT - PUT ATTRIBUTES
PUTBAK - PUT BACKGROUND ATTRIBUTES
PUTCUR - PUT CURSOR
PUTLOC - PUT LOCATION
RMVPAG - REMOVE PAGE
RPLFRM - REPLACE FORM
RSVEXP/BLDEXP - BUILD EXPRESSION TREE

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: PTHPTR/ARRAY
PURPOSE: PROCESS ARRAY
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: PTHPTR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

DESCRIPTION

PROCESS ARRAYS. IF THE SUBELEMENT IS SPECIFIED THEN ONLY THAT ELEMENT IS SEARCHED, ELSE ALL SUBELEMENTS ARE SEARCHED.

ARGUMENTS:

PATH = ELEMENT *
PASS_PATH = ELEMENT *
FLDPTR = FIELD *
LEVEL = INT
FIRST_LEV = INT

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPCODE - FORM PROCESSOR RETURN CODES
FPD - FORM PROCESSOR DATA
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

PTHPTR/FIELD - MATCH FIELD
PTHPTR/FOUND - HAS ANYTHING BEEN FOUND?

CALLED DIRECTLY BY:

PTHPTR/FIELD - MATCH FIELD

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: PTHPTR/FIELD
PURPOSE: MATCH FIELD
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: PTHPTR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

DESCRIPTION

PROCESS FIELDS BY ATTEMPTING TO MATCH THE CURRENT NAME
IN PATH WITH THE NAME IN FLDPTR. IN THIS AND THE
FOLLOWING

PROEDURES:

PATH - THE QUALIFIER TO USE AT THIS LEVEL
PASS_PATH - THE QUALIFIERS TO USE AT LOWER LEVELS
WHEN_PATH==NULL: END OF ELEMENT CHAIN
WHEN_PATH==PASS_PATH: NO QUALIFIER IN THE CHAIN ELEMENT
IS ASSOCIATED WITH THIS LEVEL
FLDPTR - POINTER TO FIELD IN THE HIERARCHY
LEVEL - THE LEVEL NUMBER OF THE FIELD INDICATED BY
FLDPTR
FIRST_LEV - THE FIRST LEVEL AT WHICH A QUALIFIER
IN THE ELEMENT CHAIN MATCHED A FIELD
NAME

ARGUMENTS:

PATH = ELEMENT *
FLDPTR = FIELD *
LEVEL = INT
FIRST_LEV = INT

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
FPCODE - FORM PROCESSOR RETURN CODES
FPD - FORM PROCESSOR DATA
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

PTHPTR/ITEM - PROCESS ITEM
PTHPTR/WINDOW - PROCESS WINDOW
PTHPTR/FORM - PROCESS FORM

MIN
PTHPTR/ARRAY - PROCESS ARRAY
STRCMP

CALLED DIRECTLY BY:

PTHPTR/ARRAY - PROCESS ARRAY
PTHPTR/FOR - PROCESS FORM
PTHPTR/WINDOW - PROCESS WINDOW
PTHPTR - GET PATH POINTER

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: PTHPTR/FORM
PURPOSE: PROCESS FORM
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: PTHPTR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

----- DESCRIPTION

PROCESS FORMS. LOOK AT ALL SUB FIELDS OF A FORM.

ARGUMENTS:

PATH = ELEMENT *
PASS_PATH = ELEMENT *
FLDPTR = FIELD *
LEVEL = INT
FIRST_LEV = INT

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
FPCODE - FORM PROCESSOR RETURN CODES
FPD - FORM PROCESSOR DATA
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

PTHPTR/FIELD - MATCH FIELD
PTHPTR/FOUND - HAS ANYTHING BEEN FOUND?

CALLED DIRECTLY BY:

PTHPTR/FIELD - MATCH FIELD

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: PTHPTR/FOUND
PURPOSE: HAS ANYTHING BEEN FOUND?
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: PTHPTR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

DESCRIPTION

DETERMINE IF A FIELD HAS BEEN LOCATED. FOR A FIELD TO BE LOCATED THE PATH NAME MUST BE THE LAST NAME ON THE ELEMENT LIST AND MATCH THE FIELD'S NAME. ITS FIRST LEVEL MUST BE LESS THAN OR EQUAL TO ANY OTHERS CURRENTLY "FOUND", IF THE FIRST LEVELS ARE EQUAL THEN IT MUST HAVE A SHORTER LENGTH PATH.

***** WARNING *****
* DUE TO AN ANOMOLY IN WINDOWS A PATH NAME THAT ENDS *
* ...WINDOW<N>; *
* WILL CAUSE PATH_PTR TO BE SET IN PROCEDURE WINDOW *
* TO THE NTH FORM IN THE WINDOW INSTEAD OF HERE IN *
* PROCEDURE FOUND. IN ADDITION LEVEL IS INCREMENTED *
* IN PROCEDURE WINDOW BEFORE THE CALL TO FOUND. *

ARGUMENTS:

PATH = ELEMENT *
PASS_PATH = ELEMENT *
FLDPTR = FIELD *
LEVEL = INT
FIRST_LEV = INT

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
FPCODE - FORM PROCESSOR RETURN CODES
FPD - FORM PROCESSOR DATA
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

STRCMP

CALLED DIRECTLY BY:

PTHPTR/ARRAY - PROCESS ARRAY
PTHPTR/FOR - PROCESS FORM
PTHPTR/ITE - PROCESS ITEM
PTHPTR/WINDOW - PROCESS WINDOW

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: PTHPTR/ITEM
PURPOSE: PROCESS ITEM
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: PTHPTR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

DESCRIPTION
PROCESS ITEMS

ARGUMENTS:

PATH = ELEMENT *
PASS_PATH = ELEMENT *
FLDPTR = FIELD *
LEVEL = INT
FIRST_LEV = INT

INCLUDE FILES:

STDTPY - STANDARD TYPE DEFINITIONS
FPCODE - FORM PROCESSOR RETURN CODES
FPD - FORM PROCESSOR DATA
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

PTHPTR/FOUND - HAS ANYTHING BEEN FOUND?

CALLED DIRECTLY BY:

PTHPTR/FIELD - MATCH FIELD

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: PTHPTR/WINDOW
PURPOSE: PROCESS WINDOW
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: PTHPTR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

DESCRIPTION

PROCESS WINDOWS. IF PAGE NUMBER IS UNSPECIFIED THE
TOP PAGE IS ASSUMED.

ARGUMENTS:

PATH = ELEMENT *
PASS_PATH = ELEMENT *
FLDPTR = FIELD *
LEVEL = INT
FIRST_LEV = INT

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPCODE - FORM PROCESSOR RETURN CODES
FPD - FORM PROCESSOR DATA
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

PTHPTR/FIELD - MATCH FIELD
PTHPTR/FOUND - HAS ANYTHING BEEN FOUND?

CALLED DIRECTLY BY:

PTHPTR/FIELD - MATCH FIELD

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: PUTATT
PURPOSE: PUT ATTRIBUTES
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FORTRAN VOID ()
SOURCE FILE: PUTATT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
FORTRAN VOID PUTATT(EWPATH, DUR, EATTRB, RCODE)
  EPATH  EWPATH;
  SHORT  *DUR;
  ENAME  EATTRB;
  CHAR   RCODE[];
```

INPUTS:

EWPATH = PATH TO FIELD POINTER CONTAINING ITEM
 CONCERNED
DUR = DURATION OF ATTRIBUTE
 (BACKGROUND/PERMINATE/TEMPORARY)
 TO BE PUT IN
EATTRB = ATTRIBUTE TO BE PUT IN

OUTPUTS:

RCODE = RETURN CODE INDICATING WHETHER OPERATION WAS
 SUCCESSFUL OR NOT (AND WHY NOT).

DESCRIPTION

THIS ROUTINE PUTS IN THE ATTRIBUTE IDENTIFIER FOR FIELDS
IF DUR IS PERM OR TEMP THEN ALL ITEMS IN WINDOW OR FORM
 SPECIFIED
OR THE ITEM ITSELF IF IT IS SPECIFIED WILL HAVE THEIR
 (ITS) PERM.
OR TEMP. ATTRIBUTE SET TO ATTRIBUTE SPECIFIED.

ARGUMENTS:

```
EWPATH =      EPATH
DUR =        SHORT *
EATTRB =      ENAME
RCODE =      CHAR []
```

INCLUDE FILES:

STDTPP - STANDARD TYPE DEFINITIONS

FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS

ROUTINES CALLED:

ESCPY - EXTERNAL STRING COPY
SYMSG - SYSTEM MESSAGE ROUTINE
PTHPTR - GET PATH POINTER
PUTATT/AABSAT - ATTRIBUTE ABSOLUTE SET ATTRIBUTE
MEMCPY
GATDEF - GET ATTRIBUTE DEFINITION

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES
UIS/FLWNST - FILL WINDOW MANAGER STRUCTURE

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: PUTATT/AABSAT
PURPOSE: ATTRIBUTE ABSOLUTE SET ATTRIBUTE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: PUTATT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

STATIC VOID AABSAT(DP, DUR, ATTRBT)

INPUTS:

DP = FIELD POINTER TO FIELD STRUCTURE TO BE
PROCESSED
DUR = DURATION OF ATTRIBUTE
(FORGROUND(PERMINATE/TEMPORARY
ATTRIBUTE)/BACKGROUND) TO BE PUT IN
ATTRB = ATTRIBUTE TO BE PUT IN

DESCRIPTION

THIS ROUTINE ABSOLUTIZES AN ATTRIBUTE NAME RELATIVE TO
THE ABSOLUTE ATTRIBUTE OF THE ELEMENT THAT CONTAINS IT
(BACKGROUND ATTRIBUTE). THE ATTRIBUTES OF ALL SUBORDINATE
NODES IN THE TREE ARE LIKEWISE ADJUSTED.

ARGUMENTS:

DP = FIELD *
DUR = SHORT *
ATTRBT = ATTMAP *

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS

ROUTINES CALLED:

PUTATT/AABSAT - ATTRIBUTE ABSOLUTE SET ATTRIBUTE
MABSAT - MAP ABSOLUTE ATTRIBUTE

CALLED DIRECTLY BY:

PUTATT/AABSAT - ATTRIBUTE ABSOLUTE SET ATTRIBUTE
PUTATT - PUT ATTRIBUTES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: PUTBAK
PURPOSE: PUT BACKGROUND ATTRIBUTES
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FORTRAN VOID ()
SOURCE FILE: PUTBAK
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
FORTRAN VOID PUTBAK(EWPATH, DUR, EATTRB, RCODE)
    EPATH  EWPATH;
    SHORT  *DUR;
    ENAME  EATTRB;
    CHAR    RCODE[];
```

INPUTS:

EWPATH = PATH TO FIELD POINTER CONTAINING ITEM
 CONCERNED
DUR = DURATION OF ATTRIBUTE (PERMINATE/TEMPORARY)
 TO BE PUT IN
EATTRB = ATTRIBUTE TO BE PUT IN

OUTPUTS:

RCODE = RETURN CODE INDICATING WHETHER OPERATION WAS
 SUCCESSFUL OR NOT (AND WHY NOT).

DESCRIPTION

THIS ROUTINE PUTS IN THE ATTRIBUTE IDENTIFIER FOR FIELDS
IF DUR IS PERM THEN ATTRIBUTE IS PUT INTO PERMINANT
 ATTRIBUTE
AND ALL FIELDS AFFECTED BY ITS ATTRIBUTE AR ADJUSTED.
 IF DUR IS
TEMP THEN IN ADDITION TO COPYING ATTRIBUTE INTO
 PERMINANT ATTRIBUTE
ETC. THE CURENT ATTRIBUTE AND CURRENT DP ARE SAVED
 BEING RESTORED
AT THE END OF OISCR.

ARGUMENTS:

EWPATH = EPATH
DUR = SHORT *
EATTRB = ENAME
RCODE = CHAR []

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS

ROUTINES CALLED:

ESCPY - EXTERNAL STRING COPY
GATDEF - GET ATTRIBUTE DEFINITION
SYSMSG - SYSTEM MESSAGE ROUTINE
MEMCPY
PTHPTR - GET PATH POINTER
RSVATT - RESOLVE ATTRIBUTE

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: PUTCUR
PURPOSE: PUT CURSOR
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FORTRAN VOID ()
SOURCE FILE: PUTCUR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

FORTRAN VOID PUTCUR(EWPATH, CODEP)

EPATH EWPATH;
CHAR CODEP[];

INPUTS:

EWPATH - PATH TO FIELD

OUTPUTS:

CODEP - RETURN CODE

DESCRIPTION

PUTCUR POSITIONS THE CURSOR AT THE SPECIFIED FIELD.

ARGUMENTS:

EWPATH = EPATH
CODEP = CHAR []

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

ESCPY - EXTERNAL STRING COPY
PTHPTR - GET PATH POINTER
POSCUR - POSITION CURSOR
MEMCPY

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES
UIS - USR INTERFACE SERVICES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: PUTLOC
PURPOSE: PUT LOCATION
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: PUTLOC
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
STATIC VOID PUTLOC(EWPATH, ROW, COL, CODEP)
    EPATH EWPATH;
    INT ROW, COL;
    CHAR CODEP[];
```

INPUTS:

EWPATH - PATH TO FIELD
ROW - ROW WITHIN THE FIELD
COL - COLUMN WITHIN THE FIELD

OUTPUTS:

CODEP - RETURN CODE

DESCRIPTION

PUTS THE CURSOR AT THE SPECIFIED ROW AND COLUMN WITHIN THE GIVEN FIELD.
THIS DIFFERS FROM THE FP PUTCUR IN THAT IT ALLOWS THE ROW AND COLUMN WITHIN A FIELD TO BE SPECIFIED. THE FP PUTCUR ALWAYS PUTS THE CURSOR AT ROW 1 COLUMN 1 WITHIN A FIELD.

ARGUMENTS:

EWPATH = EPATH
ROW = INT *
COL = INT *
CODEP = CHAR []

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

ESCPY - EXTERNAL STRING COPY
PTHPTR - GET PATH POINTER
POSCUR - POSITION CURSOR
MEMCPY

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: RMVAP
PURPOSE: REMOVE APPLICATION
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: RMVAP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
VOID RMVAP (APPTR)  
    AP *APPTR;
```

INPUTS/OUTPUTS:

INPUTS:

APPTR - AP STRUCTURE IS TO BE REMOVED

OUTPUTS:

NONE

DESCRIPTION

THIS MODULE DELETES AN AP STRUCTURE FROM INTERNAL DATA
STRUCTURE

ARGUMENTS:

APPTR = AP *

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA

ROUTINES CALLED:

SPRINTF
PMSGLS - PUT MESSAGE LINE STRING
CBIT
ATOI
RMVFPD - REMOVE FORM PROCESSOR DATA STRUCTURE
CFREE
DSPMSG

CALLED DIRECTLY BY:

MAKAP - MAKE APPLICATION STRUTURE
MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS
TRMUSR - TERMINATE USER

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: RMVFPD
PURPOSE: REMOVE FORM PROCESSOR DATA STRUCTURE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: RMVFPD
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
VOID RMVFPD(FPDPTR)  
    FPD *FPDPTR;
```

DESCRIPTION

TERMINATE FORM PROCESSOR.

ARGUMENTS:

FPDPTR = FPD *

INCLUDE FILES:

```
STDTP - STANDARD TYPE DEFINITIONS  
FPD - FORM PROCESSOR DATA  
CTLCHR - CONTROL CHARACTERS
```

ROUTINES CALLED:

```
FREMSG  
POSCUR - POSITION CURSOR  
SPRINTF  
PDVOTP - PUT DEVICE OUTPUT  
STRLEN  
DELFLD - DELETE FIELD  
CFREE
```

CALLED DIRECTLY BY:

```
CLSLDV - CLOSE LOGICAL DEVICE  
RMVAP - REMOVE APPLICATION  
RMVPD - REMOVE PHYSICAL DEVICE DATA STRUCTURE
```

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: RMVPAG
PURPOSE: REMOVE PAGE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FORTRAN VOID ()
SOURCE FILE: RMVPAG
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
FORTRAN VOID RMVPAG(EWPATH, PNUMP, RCODE)
    EPATH  EWPATH;
    INT    *PNUMP;
    CHAR   RCODE[];
```

INPUTS:

EWPATH - WINDOW TO REMOVE FROM
PNUMP - PAGE NUMBER TO REMOVE

OUTPUTS:

RCODE - RETURN CODE

DESCRIPTION

USE RMVPAG TO REMOVE A PAGE FROM A WINDOW. WHEN A PAGE IS REMOVED ALL PAGES ABOVE IT (I.E. HAVE LARGER PAGE NUMBERS) ARE ALSO REMOVED.

ARGUMENTS:

```
EWPATH = EPATH
PNUMP = INT *
RCODE = CHAR []
```

INCLUDE FILES:

```
STDTyp - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
FPPARM - FORM PROCESSOR PARAMETERS
```

ROUTINES CALLED:

```
ESCPY - EXTERNAL STRING COPY
PTHPTR - GET PATH POINTER
SYSMSG - SYSTEM MESSAGE ROUTINE
```

DELFLD - DELETE FIELD
MAX
STRASN
MEMCPY

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES
UIS - USR INTERFACE SERVICES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: RMVPD
PURPOSE: REMOVE PHYSICAL DEVICE DATA STRUCTURE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: RMVPD
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
VOID RMVPD(PDPTR)
    PD *PDPTR;
    INT FLG;
```

INPUTS/OUTPUTS:

INPUTS:

PDPTR - PD STRUCTURE IS TO BE REMOVED
FLG - 0 = DO NOT REMOVE LOGICAL DEVICE(FPD) - NRMVFPD
1 = REMOVE LOGICAL DEVICE(FPD) AND CALL TRMDRV
- RMVFPD
2 = DO NOT CALL TRMDRV DEVICE ALREADY DEAD -
DEAD

OUTPUTS:

NONE

DESCRIPTION

THIS MODULE DELETES AN PD STRUCTURE FROM INTERNAL DATA
STRUCTURE

ARGUMENTS:

PDPTR = PD *
FLG = INT

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

FCLOSE
RMVFPD - REMOVE FORM PROCESSOR DATA STRUCTURE
TRMDRV - TERMINATE DEVICE DRIVER
CFREE

CALLED DIRECTLY BY:

MAKUSR - MAKE USER
MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS
TRMUSR - TERMINATE USER
UIS/STRTAP - START APPLICATION
UIS/PRCINP - PROCESS INPUT

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: RPLFRM
PURPOSE: REPLACE FORM
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FORTRAN VOID ()
SOURCE FILE: RPLFRM
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
FORTRAN VOID RPLFRM(EWPATH, PNUMP, EFNAME, RCODE)
    EPATH EWPATH;
    INT *PNUMP;
    ENAME EFNAME;
    CHAR RCODE[];
```

INPUTS:

EWPATH - PATH OF WINDOW TO REPLACE IN
PNUMP - PAGE NUMBER TO REPLACE
EFNAME - FORM TO REPLACE WITH

OUTPUTS:

RCODE - RETURN CODE

DESCRIPTION

USE RPLFRM TO REPLACE A FORM IN A WINDOW.

ARGUMENTS:

```
EWPATH = EPATH
PNUMP = INT *
EFNAME = ENAME
RCODE = CHAR []
```

INCLUDE FILES:

```
STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
```

ROUTINES CALLED:

```
ESCPY - EXTERNAL STRING COPY
PTHPTR - GET PATH POINTER
SYSMSG - SYSTEM MESSAGE ROUTINE
```


DELFLD - DELETE FIELD
COPFRM - COPY FORM
RSVATT - RESOLVE ATTRIBUTE
MEMCPY

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: RSVATT
PURPOSE: RESOLVE ATTRIBUTE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: RSVATT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

VOID RSVATT(DP)
FIELD *DP

INPUTS:

*DP - FIELD POINTER OF FIELD WHOSE ATTRIBUTE IS BEING
RESOVED.

DESCRIPTION

RSVATT RESOLVES BACKGROUND ATTRIBUTES

ARGUMENTS:

DP = FIELD *

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA

ROUTINES CALLED:

RSVATT/RSVRST - RESOLVE REST
STRASN

CALLED DIRECTLY BY:

COPFLD - COPY FIELD
PUTBAK - PUT BACKGROUND ATTRIBUTES
RPLFRM - REPLACE FORM

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: RSVATT/RSVRST
PURPOSE: RESOLVE REST
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: RSVATT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

STATIC VOID RSVRST(DP)
FIELD *DP;

INPUTS:

*DP - FIELD POINTER OF 1RST FIELD IN LEVEL BEING
RESOVED.

DESCRIPTION

RSVRST DOES THE REAL WORK OF RESOLVING BACKGROUND
ATTRIBUTES

ARGUMENTS:

DP = FIELD *

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA

ROUTINES CALLED:

RSVATT/RSVRST - RESOLVE REST
STRASN
MABSAT - MAP ABSOLUTE ATTRIBUTE

CALLED DIRECTLY BY:

RSVATT/RSVRST - RESOLVE REST
RSVATT - RESOLVE ATTRIBUTE

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: RSVEXP
PURPOSE: RESOLVE EXPRESSIONS
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: RSVEXP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

CHAR *RSVEXP(DP)
FIELD *DP;

INPUTS:

DP - POINTER TO FIELD TO RESOLVE

DESCRIPTION

ALL ITEMS SUBORDINATE TO THE GIVEN FIELD ARE EXAMINED FOR
AN EXPRESSION
WHICH IS BUILT INTO A TREE AND EVALUATED.

ARGUMENTS:

DP = FIELD *

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

RSVEXP/BLDEXP - BUILD EXPRESSION TREE
SYSMSG - SYSTEM MESSAGE ROUTINE
RSVEXP - RESOLVE EXPRESSIONS
CMPFLD - COMPUTE FIELD

CALLED DIRECTLY BY:

ADDELM - ADD ELEMENT
COPFRM - COPY FORM
RSVEXP - RESOLVE EXPRESSIONS

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: RSVEXP/BLDEXP
PURPOSE: BUILD EXPRESSION TREE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: RSVEXP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

----- SYNOPSIS

```
STATIC CHAR *BLDEXP(DP, S, EPP)
FIELD *DP;
CHAR **S;
ENODE **EPP;
```

INPUTS:

DP - POINTER TO FIELD CONTAINING THE EXPRESSION
S - EXPRESSION STRING TO BUILD TREE FROM

OUTPUTS:

EPP - POINTER TO ROOT OF BUILT TREE
RETURNS AN ERROR CODE OR NULL

DESCRIPTION

SETS A POINTER TO THE ROOT OF THE EXPRESSION TREE BUILT
FROM THE SUPPLIED
STRING.

ARGUMENTS:

DP = FIELD *
SP = CHAR **
EPP = ENODE **

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
CTYPE - **** PURPOSE NOT FOUND BY STRIPPER ****
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

FREE
RSVEXP/BLDEXP - BUILD EXPRESSION TREE
ISDIGIT

STRCHR
PTHPTR - GET PATH POINTER
SYMSG - SYSTEM MESSAGE ROUTINE
MALLOC

CALLED DIRECTLY BY:

RSVEXP/BLDEXP - BUILD EXPRESSION TREE
RSVEXP - RESOLVE EXPRESSIONS

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: SFPDAP
PURPOSE: SET FORM PROCESSOR DATA STRUCTURE FOR
APPLICATION
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: SFPDAP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
CHAR *SFPDAP (APNAM, APCHAN)
    NAME APNAM;
    CHAN APCHAN;
```

INPUTS/OUTPUTS:

INPUTS:

APNAM - NAME OF AP STRUCTURE WILL BE FOR - FROM NTM
APCHAN - UNIQUE INSTANCE OF AP - FROM NTM

OUTPUTS:

RETURNS POINTER TO ERROR CODE FOR FORM PROCESSOR IF
ERROR
OR A NULL POINTER

DESCRIPTION

THIS MODULE IS CALLED BY MONITOR ROUTINE TO SET UP THE FPD
STRUCTURE INORDER TO MAKE ITS CALL TO "CALLFP".

ARGUMENTS:

APNAM =	ENAME
APCHAN =	CHAN

INCLUDE FILES:

STDYTP	- STANDARD TYPE DEFINITIONS
FPD	- FORM PROCESSOR DATA
FPCODE	- FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

ESCPY - EXTERNAL STRING COPY
STRCMP
SYSMSG - SYSTEM MESSAGE ROUTINE

CALLED DIRECTLY BY:

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: STUPFP
PURPOSE: SET UP FORM PROCESSOR DATA STRUCTURES
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: STUPFP
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
CHAR *STUPFP(FPDPTR)
    FPD *FPDPTR;
```

INPUTS/OUTPUTS:

INPUTS:

FPDPTR - POINTER TO LOGICAL DEVICE(FPD)

OUTPUTS:

RETURNS A POINTER TO ERROR CODE IF ERRRO OR A NULL
POINTER NO ERROR

DESCRIPTION

OPENS PSCREEN FORM, CREATES FIRST WINDOW AND COPIES THIS
STUCTURE INTO
FIELD DISPLAY LIST (FIRST ENTERY).

ARGUMENTS:

FPDPTR = FPD *

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

GATDEF - GET ATTRIBUTE DEFINITION
MAKFLD - MAKE FIELD
COPFRM - COPY FORM
SYSMSG - SYSTEM MESSAGE ROUTINE

SBIT
STRASN
FFBCA

CALLED DIRECTLY BY:

MAKAP - MAKE APPLICATION STRUTURE
MAKUSR - MAKE USER
OPNLDV - OPEN LOGICAL DEVICE

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: SYMSG
PURPOSE: SYSTEM MESSAGE ROUTINE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: CHAR * ()
SOURCE FILE: SYMSG
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
CHAR *SYMSG(CODE)
CHAR *CODE;
```

INPUTS:

CODE - ERROR CODE

DESCRIPTION

IF ERROR CODE IS FATAL TO FORM PROCESSOR (79,000 ><
79,999), IT
GETS ERROR CODE AND ERROR MESSAGES AND PASSES THEM TO
ERRPRO()
WHICH WRITES THEM TO AN ERROR LOG FILE

ARGUMENTS:

CODE = CHAR *

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

atoi
FNDMSG - FIND MESSAGE
ERRPRO
STRNCMP
PMSGCL - PUT MESSAGE LINE CODE

CALLED DIRECTLY BY:

ADDELM - ADD ELEMENT

ADDFRM - ADD FORM TO WINDOW
CALLFP - CALL FP ROUTINES
CLSFRM - CLOSE FORM
CMPFLD/EVA - EVALUATE FIELD EXPRESSION
CMPFLD - COMPUTE FIELD
COPFLD/CPYFLD - INTERNAL COPY FIELD
COPFRM - COPY FORM
DELFLD - DELETE FIELD
GATDEF - GET ATTRIBUTE DEFINITION
GDATA - GET DATA
GDVINP - GET DEVICE INPUT
GETATT - GET ATTRIBUTE
GETBAK - GET BACKGROUND ATTRIBUTE
GETCUR - GET CURSOR POSITION
GPAGE - GET PAGE
MAKAP - MAKE APPLICATION STRUCTURE
MAKFLD - MAKE FIELD
MAKUSR - MAKE USER
MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS
OISCR/CNGMSG - CHECK FOR AND PROCESS CHANGE MESSAGE REQUESTS
OISCR/PROCFLD - PROCESS FIELD
OISCR/EVTBUF - EMPTY VTI BUFFER
ONWISC - OUTPUT (NO WAIT) / INPUT SCREEN
INSCR - INPUT SCREEN
OPNFRM/PFREC - PROCESS FORM RECORD
OPNFRM/PTREC - PROCESS TEXT RECORD
OPNFRM/PDREC - PROCESS FIELD RECORD
OPNFRM/PAR - PROCESS ARRAY
OPNFRM/PIT - PROCESS ITEM
OPNFRM/PFR - PROCESS FORM
OPNFRM/BTBUF - BUILD TEXT BUFFER
OPNFRM/BDBUFF - BUILD DEFAULT BUFFER
OPNFRM/BFLDDB - BUILD FIELD DEFAULT BUFFER
OPNFRM/BRPNOD - BUILD RELATIVE POSITION NODE
OPNFRM - OPEN FORM
PARFQN - PARSE FULLY QUALIFIED NAME
PDATA/PUTBUF - PUT BUFFER
PDVOTP - PUT DEVICE OUTPUT
PTHPTR - GET PATH POINTER
PUTATT - PUT ATTRIBUTES
PUTBAK - PUT BACKGROUND ATTRIBUTES
RMVPAG - REMOVE PAGE
RPLFRM - REPLACE FORM
RSVEXP/BLDEXP - BUILD EXPRESSION TREE
RSVEXP - RESOLVE EXPRESSIONS
SFPDAP - SET FORM PROCESSOR DATA STRUCTURE FOR APPLICATION
STUPFP - SET UP FORM PROCESSOR DATA STRUCTURES
TRMDRV - TERMINATE DEVICE DRIVER
TRMUSR - TERMINATE USER
UIS/STRTAP - START APPLICATION
UIS/STRTPD - START PHYSICAL DEVICE
UIS/PRCINP - PROCESS INPUT
UIS - USER INTERFACE SERVICES

PS 620344200
30 September 1990

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: TERMVT
PURPOSE: TERMINATE VIRTUAL TERMINAL INTERFACE
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: TERMVT
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

TERMVT(RCODE)
CHAR RCODE[]

INPUTS/OUTPUTS:

INPUTS:
NONE

OUTPUTS:
RCODE - STANDARD FORM PROCESSOR RETURN CODE

DESCRIPTION

CLEARs VTI MODE FLAG AND SET MAX BUFF LENGTH BACK TO 0

ARGUMENTS:

RCODE = CHAR []

INCLUDE FILES:

STDTPY - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES

ROUTINES CALLED:

MEMCPY

CALLED DIRECTLY BY:

CALLFP - CALL FP ROUTINES

PS 620344200
30 September 1990

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: TRMDRV
PURPOSE: TERMINATE DEVICE DRIVER
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: TRMDRV
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
VOID TRMDRV(PDPTR)
    PD          *PDPTR;
```

INPUTS/OUTPUTS:

INPUTS:

PDPTR - POINTER TO DEVICE DRIVER TERMINATING

OUTPUTS:

NONE

DESCRIPTION

THIS MODULE SENDS A SHUT DOWN MESSAGE TO DEVICE DRIVER
POINTED TO

ARGUMENTS:

PDPTR = PD *

INCLUDE FILES:

STDTP	- STANDARD TYPE DEFINITIONS
FPD	- FORM PROCESSOR DATA
FPCODE	- FORM PROCESSOR RETURN CODES
NTM	- NTM INTERFACE INCLUDE FILE

ROUTINES CALLED:

MEMSET
MEMCPY
STRLEN

NSEND
MEMCMP
SYSMSG - SYSTEM MESSAGE ROUTINE

CALLED DIRECTLY BY:

RMVPD - REMOVE PHYSICAL DEVICE DATA STRUCTURE

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: TRMUSR
PURPOSE: TERMINATE USER
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: TRMUSR
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
VOID TRMUSR(USRPTR)
    USR *USRPTR;
```

INPUTS/OUTPUTS:

INPUTS:

USRPTR - USER STRUCTURE IS TO BE REMOVED

OUTPUTS:

NONE

DESCRIPTION

THIS MODULE DELETES A USER STRUCTURE FROM INTERNAL DATA
STRUCTURE
AND SENDS TERMINATING MESSAGES TO ALL THE USER'S PROCESSES

ARGUMENTS:

USRPTR = USR *

INCLUDE FILES:

STDTPY - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPCODE - FORM PROCESSOR RETURN CODES
NTM - NTM INTERFACE INCLUDE FILE

ROUTINES CALLED:

MEMSET
MEMCPY
STRLEN
SIGABT
MEMCMP

SYSMSG - SYSTEM MESSAGE ROUTINE
RMVAP - REMOVE APPLICATION
RMVPD - REMOVE PHYSICAL DEVICE DATA STRUCTURE
CFREE

CALLED DIRECTLY BY:

MAKUSR - MAKE USER
MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS
UIS - USR INTERFACE SERVICES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: UIS
PURPOSE: USR INTERFACE SERVICES
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: UIS
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS
UIS()

INPUTS/OUTPUTS:

INPUTS:
NONE

OUTPUTS:
NONE

DESCRIPTION

THIS MODULE CONTAINES THE USER INTERFACE LOGON, FUNCTION
PICK
AND WINDOW MANAGER SERVICE.

INCLUDE FILES:

STDTyp - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES
CTLCHR - CONTROL CHARACTERS
DBASEI - DATABASE INTERFACE
NTM - NTM INTERFACE INCLUDE FILE
UISFM - UIS FORM

ROUTINES CALLED:

OPNFRM - OPEN FORM
MEMCMP
UIS/PRCWND - PRCESS WINDOW
UIS/FLWNST - FILL WINDOW MANAGER STRUCTURE
PMSGLC - PUT MESSAGE LINE CODE

DBCROL	- CHECK ROLE
PUTCUR	- PUT CURSOR
MEMSET	
MEMCPY	
UIS/STRTAP	- START APPLICATION
PDATA	- PUT FORM DATA
ESCPY	- EXTERNAL STRING COPY
SPRINTF	
PDVOTP	- PUT DEVICE OUTPUT
STRLEN	
ONWISC	- OUTPUT (NO WAIT) / INPUT SCREEN
SYSMSG	- SYSTEM MESSAGE ROUTINE
TRMUSR	- TERMINATE USER
OUTSCR	- OUTPUT SCREEN
PMSGLS	- PUT MESSAGE LINE STRING
RMVPAG	- REMOVE PAGE
DBCUPR	
GDATA	- GET DATA
ADDFRM	- ADD FORM TO WINDOW

CALLED DIRECTLY BY:

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: UIS/FLWINF
PURPOSE: FILL WINDOW INFORMATION
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: UIS
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
STATIC VOID FLWINF(WNMNGR, APNAM, APCHAN, DP, LEV)
    STRUCT WNMNGR *WNMNGR;
    NAME          APNAM;
    CHAN          APCHAN;
    FIELD         *DP;
    INT           *I;
```

INPUTS/OUTPUTS:

INPUTS:

WNMNGR - DATA STRUCTURE FOR WINDOW MANAGER WHICH IS TO
 BE FILLED
APNAM - NAME OF APPLICATION TO WHICH WINDOW BELONGS
APCHAN - CHANNEL OF APPLICATION TO WHICH WINDOW BELONGS
DP - POINTER TO CURRENT FIELD
INT - INDEX ARRAY OF DATA STRUCTURES

OUTPUTS:

NONE

DESCRIPTION

THIS MODULE FILLS DATA STRUCTURE FOR PARTICULAR WINDOW
FROM FPD STRUCTURE

ARGUMENTS:

```
WNMNGR = STRUCT WNMNGR *
APNAM = NAME
APCHAN = CHAN
DP = FIELD *
I = INT *
```

INCLUDE FILES:

```
STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
```

FPCODE	- FORM PROCESSOR RETURN CODES
CTLCHR	- CONTROL CHARACTERS
DBASEI	- DATABASE INTERFACE
NTM	- NTM INTERFACE INCLUDE FILE
UISFM	- UIS FORM

ROUTINES CALLED:

UIS/FLWINF - FILL WINDOW INFORMATION
SPRINTF

CALLED DIRECTLY BY:

UIS/FLWNST - FILL WINDOW MANAGER STRUCTURE
UIS/FLWINF - FILL WINDOW INFORMATION

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: UIS/FLWNST
PURPOSE: FILL WINDOW MANAGER STRUCTURE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: BOOL ()
SOURCE FILE: UIS
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

STATIC BOOL FLWNST(WNMGR)
STRUCT WNMGR *WNMGR;

INPUTS/OUTPUTS:

INPUTS:

WNMGR - DATA STRUCTURE FOR WINDOW MANAGER WHICH IS TO
BE FILLED

OUTPUTS:

RETURNS SUCCESS/FAILURE

DESCRIPTION

THIS MODULE FILLS DATA STRUCTURE FOR WINDOW MANAGER FROM
FPD STRUCTURE

ARGUMENTS:

WNMGR = STRUCT WNMGR *

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES
CTLCHR - CONTROL CHARACTERS
DBASEI - DATABASE INTERFACE
NTM - NTM INTERFACE INCLUDE FILE
UISFM - UIS FORM

ROUTINES CALLED:

UIS/FLWINF - FILL WINDOW INFORMATION
ESCPY - EXTERNAL STRING COPY

MEMCMP
PUTATT - PUT ATTRIBUTES
SPRINTF
MEMSET

CALLED DIRECTLY BY:

UIS - USR INTERFACE SERVICES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: UIS/PRCINP
PURPOSE: PRCESS INPUT
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: FPD * ()
SOURCE FILE: UIS
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
STATIC FPD *PRCINP(WNINP, OWNINP)
    STRUCT WNDMNGINP *WNINP
    STRUCT WNDMNGINP *OWNINP;
```

INPUTS/OUTPUTS:

INPUTS:

WNINP - DATA STURCTURE FOR WINDOW MANAGER WHICH IS TO
BE PROCESSSED
OWNINP - OLD DATA STURCTURE - USED TO SEE IF CHANGES
WERE MADE

OUTPUTS:

RETURNS POINTER TO FPD STRUCTURE IF SUCCESS ELSE A NULL

DESCRIPTION

THIS MODULE PROCESSES WINDOW MANAGER INPUT AND MAKES THE
APPROPRIATE
CHANGES TO FPD DATA STRUCTURE.

ARGUMENTS:

WNINP = STRUCT WNDMNGINP *
OWNINP = STRUCT WNDMNGINP *

INCLUDE FILES:

STDYTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES
CTLCHR - CONTROL CHARACTERS
DBASEI - DATABASE INTERFACE
NTM - NTM INTERFACE INCLUDE FILE
UISFM - UIS FORM

ROUTINES CALLED:

MAX
MIN
ULKFPD - UNLINKK FPD
RMVPD - REMOVE PHYSICAL DEVICE DATA STRUCTURE
STRLEN
PDVOTP - PUT DEVICE OUTPUT
SPRINTF
UIS/STRTPD - START PHYSICAL DEVICE
SYSMSG - SYSTEM MESSAGE ROUTINE
STRCMP
ESCPY - EXTERNAL STRING COPY
FNFPWN - FIND FORM PROCESSOR WINDOW
MEMCMP
MATOI

CALLED DIRECTLY BY:

UIS/PRCWND - PRCESS WINDOW

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: UIS/PRCWND
PURPOSE: PRCESS WINDOW
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: BOOL ()
SOURCE FILE: UIS
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
STATIC VOID PRCWND(WNMNGR, OWMNNGR)
    STRUCT WNMNGR *WNMNGR;
    STRUCT WNMNGR *OWNMNGR;
```

INPUTS/OUTPUTS:

INPUTS:

WNMNGR - DATA STURCTURE FOR WINDOW MANAGER WHICH IS TO
BE FILLED

OUTPUTS:

RETURNS SUCCESS/FAILURE

DESCRIPTION

THIS MODULE PROCESSES WINDOW MANAGER INPUT AND MAKES THE
APPROPRIATE
CHANGES TO FPD DATA STRUCTURE.

ARGUMENTS:

```
WNMNGR = STRUCT WNMNGR *
OWNMNGR = STRUCT WNMNGR *
```

INCLUDE FILES:

```
STDTyp - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES
CTLCHR - CONTROL CHARACTERS
DBASEI - DATABASE INTERFACE
NTM - NTM INTERFACE INCLUDE FILE
UISFM - UIS FORM
```

ROUTINES CALLED:

CHGPRC - CHANGE PRECEDENCE OF WINDOW OR LOGICAL DEVICE
PMSGSL - PUT MESSAGE LINE STRING
SPRINTF
MATOI
UIS/PRCINP - PRCESS INPUT
MEMCMP
MEMSET

CALLED DIRECTLY BY:

UIS - USR INTERFACE SERVICES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: UIS/STRTPAP
PURPOSE: START APPLICATION
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: UIS
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

STATIC VOID STRTPAP(FUNCTN)
STRUCT FUNCTN *FUNCTN;

INPUTS/OUTPUTS:

INPUTS:

FUNCTN - INPUT DATA FROM THE FUNCTION SCREEN

OUTPUTS:

NONE

DESCRIPTION

THIS MODULE STARTS UP AN APPLICATION AFTER MAKING SURE
USER ALLOWED TO
RUN IT. IT ALSO CALLS MAKAP TO CREATE THE DATA STRUCTURE
FOR THE
APPLICATION AND STRTPD TO START THE PHYSICAL DEVICE IF
SPECIFIED

ARGUMENTS:

FUNCTN = STRUCT FUNCTN *

INCLUDE FILES:

STDYTP	- STANDARD TYPE DEFINITIONS
FPD	- FORM PROCESSOR DATA
FPPARM	- FORM PROCESSOR PARAMETERS
FPCODE	- FORM PROCESSOR RETURN CODES
CTLCHR	- CONTROL CHARACTERS
DBASEI	- DATABASE INTERFACE
NTM	- NTM INTERFACE INCLUDE FILE
UISFM	- UIS FORM

ROUTINES CALLED:

SYSMSG - SYSTEM MESSAGE ROUTINE
SIGABT
MAKAP - MAKE APPLICATION STRUTURE
ISEND
RMVPD - REMOVE PHYSICAL DEVICE DATA STRUCTRUE
PMSGLS - PUT MESSAGE LINE STRING
SPRINTF
FFBCA
DBCOM
STRLEN
ESCPY - EXTERNAL STRING COPY
DBGAPD
UIS/STRTPD - START PHYSICAL DEVICE
PMSGLC - PUT MESSAGE LINE CODE
DBCFCNC - CHECK FUNCTION
MEMCMP

CALLED DIRECTLY BY:

UIS - USR INTERFACE SERVICES

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: UIS/STRTPD
PURPOSE: START PHYSICAL DEVICE
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: PD * ()
SOURCE FILE: UIS
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

----- SYNOPSIS

```
    STATIC PD *STRTPD(DEVICE, DEVTYP)
           ENAME      DEVICE;
           ENAME      DEVTYP;
```

INPUTS/OUTPUTS:

INPUTS:

DEVICE - NAME OF ACTUAL PHYSICAL DEVICE
DEVTYP - NAME OF DEVICE DRIVER

OUTPUTS:

RETURNS POINTER TO (NEW OR CURRENT) PHYSICAL DEVICE OR
NULL IF FAILED

DESCRIPTION

THIS MODULE STARTS UP AN PHYSICAL DEVICE AFTER MAKPD TO
CREATE THE DATA
STRUCTURE FOR THE DEVICE.

ARGUMENTS:

DEVICE = ENAME
DEVTYP = ENAME

INCLUDE FILES:

STDTP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA
FPPARM - FORM PROCESSOR PARAMETERS
FPCODE - FORM PROCESSOR RETURN CODES
CTLCHR - CONTROL CHARACTERS
DBASEI - DATABASE INTERFACE
NTM - NTM INTERFACE INCLUDE FILE
UISFM - UIS FORM

ROUTINES CALLED:

SYMSG - SYSTEM MESSAGE ROUTINE
SIGABT
MAKPD - MAKE PHYSICAL DEVICE STRUCTURE
ISEND
FFBCA
STRLEN
FCLOSE
FOPEN
PMSGLS - PUT MESSAGE LINE STRING
SPRINTF
FREE
FSEARCH
STRCMP
ESCPY - EXTERNAL STRING COPY
MEMCMP

CALLED DIRECTLY BY:

UIS/STRTAP - START APPLICATION
UIS/PRCINP - PRCESS INPUT

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

FORM PROCESSOR Module Documentation

NAME: ULKFPD
PURPOSE: UNLINKKK FPD
LANGUAGE: C
MODULE TYPE: SUBROUTINE
FUNCTION TYPE: VOID ()
SOURCE FILE: ULKFPD
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: UI
SUBDIRECTORY: FP
DOCUMENTATION GROUP: FORMPROC

DESCRIPTION:

SYNOPSIS

```
VOID ULKFPD(FPDPTR)  
    FPD *FPDPTR;
```

INPUTS/OUTPUTS:

INPUTS:

FPDPTR - FPD STRUCTURE IS TO BE UNLINKED

OUTPUTS:

NONE

DESCRIPTION

THIS MODULE UNLINKS FPD STRUCTURE FROM PD STRUCTURE

ARGUMENTS:

FPDPTR = FPD *

INCLUDE FILES:

STD TYP - STANDARD TYPE DEFINITIONS
FPD - FORM PROCESSOR DATA

CALLED DIRECTLY BY:

UIS/PRCINP - PRCESS INPUT

USED IN MAIN PROGRAM(S):

MONITR/MAI - MAIN MODULE FOR MONITOR/UIS/FP PROCESS

3.10.9 Include File Descriptions

The following list contains a purpose and description of each include file listed in 3.10.4 as specified in the source code. The language it is written in is also given.

FORM PROCESSOR Include File Description

FILE NAME: BITS
PURPOSE: INCLUDE FILE FOR BIT MANIPULATION ROUTINES
LANGUAGE: C

DESCRIPTION:

FORM PROCESSOR Include File Description

FILE NAME: CICODE
PURPOSE: Command Interpreter CODEs
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

IDENTIFICATION: CICODE

DESCRIPTION:

THESE ARE COMMAND INTERPRETER CODES.

INFORMATION:

TYPE: (C-COBOL, IC-COBOL COPY) IC
SUBSYSTEM: UI-CI
CONFIGURATION ITEM ID:

DESIGNED BY: S. L. BARKER
START DATE: 1/18/83
FINISH DATE: 1/18/83

PROGRAMMED BY: S. L. BARKER
START DATE: 1/18/83
FINISH DATE: 1/18/83

UPDATED 8/24/83 TO COMBINE WITH UICODE.INC
UPDATED 8/25/83 TO ACCOMMODATE NEW MESSAGE LINE CODE

FORM PROCESSOR Include File Description

FILE NAME: CTLCHR
PURPOSE: CONTROL CHARACTERS
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
DEFINITIONS OF ALL CONTROL CHARACTERS TO AVOID CHARACTER
SET
DEPENDENCIES.

FORM PROCESSOR Include File Description

FILE NAME: CURSORI
PURPOSE: CURSOR description
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

IDENTIFICATION: CURSOR

DESCRIPTION:

THIS IS THE ORACLE CURSOR DESCRIPTION.
INFORMATION:

TYPE: (C-COBOL, IC-COBOL COPY) IC
SUBSYSTEM: UI
CONFIGURATION ITEM ID:

DESIGNED BY: S. L. BARKER

FORM PROCESSOR Include File Description

FILE NAME: DBASEI
PURPOSE: DATABASE ITERFACE
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
INCLUDE FILE FOR DATA BASE INTERFACE CALLS

FORM PROCESSOR Include File Description

FILE NAME: FFFV2
PURPOSE: FORM FILE FORMAT - VERSION 2
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
RECORD LAYOUTS FOR THE BINARY FORM DEFINITION FILE

FORM PROCESSOR Include File Description

FILE NAME: FPCODE
PURPOSE: FORM PROCESSOR RETURN CODES
LANGUAGE: C

DESCRIPTION:

FORM PROCESSOR Include File Description

FILE NAME: FPD
PURPOSE: FORM PROCESSOR DATA
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
DATA DEFINITIONS FOR ALL FORM PROCESSOR (INCLUDING
MONITER) DATA.

FORM PROCESSOR Include File Description

FILE NAME: FPDINI
PURPOSE: FPD INITIALIZATION
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
INITIALIZED VERSION OF UID FOR INCLUSION IN MAIN PROGRAM.

FORM PROCESSOR Include File Description

FILE NAME: FPEMSG
PURPOSE: FORM PROCESSOR ERROR MESSAGES
LANGUAGE: C

DESCRIPTION:

DESCRIPTION

FORM PROCESSOR Include File Description

FILE NAME: FPPARM
PURPOSE: FORM PROCESSOR PARAMETERS
LANGUAGE: C

DESCRIPTION:

DESCRIPTION: THESE DATA DEFINITIONS ARE USED
IN THE FORM PROCESSOR ROUTINES.

FORM PROCESSOR Include File Description

FILE NAME: FUNCTS
PURPOSE: FUNCTION DEFINITIONS
LANGUAGE: C

DESCRIPTION:

DESCRIPTION

DEFINES THE MNEMONIC VIRTUAL TERMINAL COMMAND FUNCTIONS.
AND DEFINES STRUCTURE FOR PARSING VTI MESSAGE BUFFER.

FORM PROCESSOR Include File Description

FILE NAME: NTM
PURPOSE: NTM INTERFACE INCLUDE FILE
LANGUAGE: C

DESCRIPTION:

DESCRIPTION
INCLUDE FILE FOR NTM INTERFACE

FORM PROCESSOR Include File Description

FILE NAME: ORACLE
PURPOSE: data delcarations for programs that access ORACLE
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

DESCRIPTION:

THESE ARE DATA DECLARATIONS THAT ARE COMMONLY USED
IN PROGRAMS THAT ACCESS ORACLE.

INFORMATION:

TYPE: (C-COBOL, IC-COBOL COPY) IC
SUBSYSTEM: UI
CONFIGURATION ITEM ID:

DESIGNED BY: S. L. BARKER
START DATE: 1/17/83
FINISH DATE: 1/17/83

PROGRAMMED BY: S. L. BARKER
START DATE: 1/17/83
FINISH DATE: 1/24/83

FORM PROCESSOR Include File Description

FILE NAME: ORCODE
PURPOSE: ORacle CODEs
LANGUAGE: VAX-11 COBOL

DESCRIPTION:

FORM PROCESSOR Include File Description

FILE NAME: STD TYP
PURPOSE: STANDARD TYPE DEFINITIONS
LANGUAGE: C

DESCRIPTION:

DESCRIPTION

THIS FILE ENSURES THAT THE FOLLOWING STANDARD TYPES ARE
AVAILABLE:

FLOAT	- SINGLE PRECISION FLOAT
DOUBLE	- DOUBLE PRECISION FLOAT
LONG	- 32 BIT (OR LARGER) SIGNED INTEGER
LBITS	- 32 BITS (OR MORE) FOR BIT MANIPULATION
INT	- NATURAL SIZE SIGNED INTEGER
UNSIGNED	- NATURAL SIZE UNSIGNED INTEGER
BOOL	- NATURAL SIZE LOGICAL (ZERO / NON-ZERO ONLY)
SHORT	- 16 BIT (OR LARGER) SIGNED INTEGER
USHORT	- 16 BIT (OR LARGER) UNSIGNED INTEGER
BITS	- 16 BITS (OR MORE) FOR BIT MANIPULATION
CHAR	- SINGLE MACHINE CHARACTER (REAL CHARACTERS ALWAYS POSITIVE)
TINY	- 8 BIT (OR LARGER) SIGNED INTEGER
UTINY	- 8 BIT (OR LARGER) UNSIGNED INTEGER
TBITS	- 8 BITS (OR MORE) FOR BIT MANIPULATION
TBOOL	- 8 BIT (OR LARGER) LOGICAL (ZERO / NON-ZERO ONLY)
METACHAR	- 16 BIT (OR LARGER) AUGMENTED CHARACTER (SIGNED)
VOID	- FUNCTION THAT RETURNS NO VALUE
FORTTRAN	- STORAGE CLASS FOR FOREIGN (NON-C) ROUTINES OR C ROUTINES WHICH ARE CALLABLE FROM FOREIGN ROUTINES

SINCE NOT ALL COMPILERS SUPPORT USHORT, TINY, AND UTINY,
THE FUNCTIONS
USHORT(), TINY(), AND UTINY() SHOULD BE USED WHENEVER
REFERENCING THEM.

IN ADDITION, THE FOLLOWING UTILITY MACROS ARE DEFINED:
LURSHIFT(N, B) - UNSIGNED LONG RIGHT SHIFT
MAX(A, B) - MAXIMUM OF A AND B
MIN(A, B) - MINIMUM OF A AND B

FORM PROCESSOR Include File Description

ABS(A) - ABSOLUTE VALUE OF A
STRASN(A, B) - TRANSPORTABLE A = B FOR STRUCTURES
NULL - NULL POINTER VALUE (0)
TRUE - 1
FALSE - 0
SUCCESS - EXIT(SUCCESS) INDICATES SUCCESSFUL
COMPLETION
FAILURE - EXIT(FAILURE) INDICATES ERRORS

THE FOLLOWING SYMBOLS SHOULD BE DEFINED BASED ON THE
COMPILER BEING USED:

USHORT - COMPILER SUPPORTS UNSIGNED SHORT
TINY - COMPILER TREATS CHAR AS SIGNED
UTINY - CHAR IS SIGNED AND COMPILER SUPPORTS
UNSIGNED CHAR
VOID - COMPILER SUPPORTS VOID
FORTRAN - COMPILER SUPPORTS FORTRAN
STRASN - DEFINE APPROPRIATE MACRO
SUCCESS - DEFINE APPROPRIATE VALUE IF NOT 0
FAILURE - DEFINE APPROPRIATE VALUE IF NOT 1

FORM PROCESSOR Include File Description

FILE NAME: UISFM
PURPOSE: UIS FORM
LANGUAGE: C

DESCRIPTION:

FORM PROCESSOR Include File Description

FILE NAME: VTICOM
PURPOSE: VTI COMMUNICATION DEFINITIONS
LANGUAGE: C

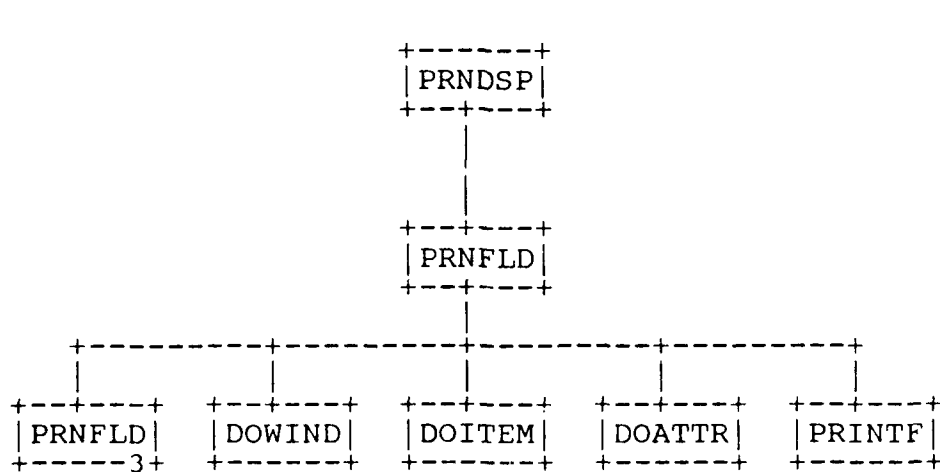
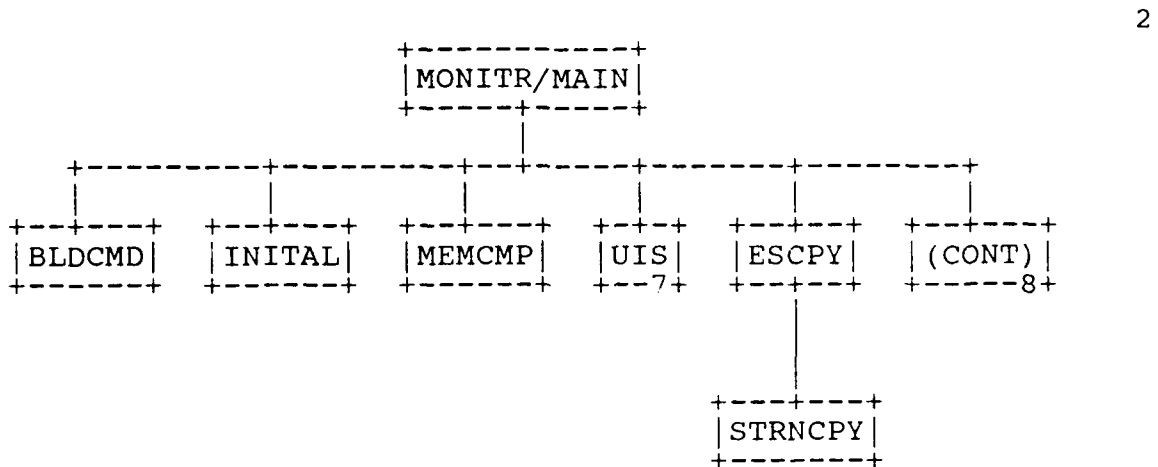
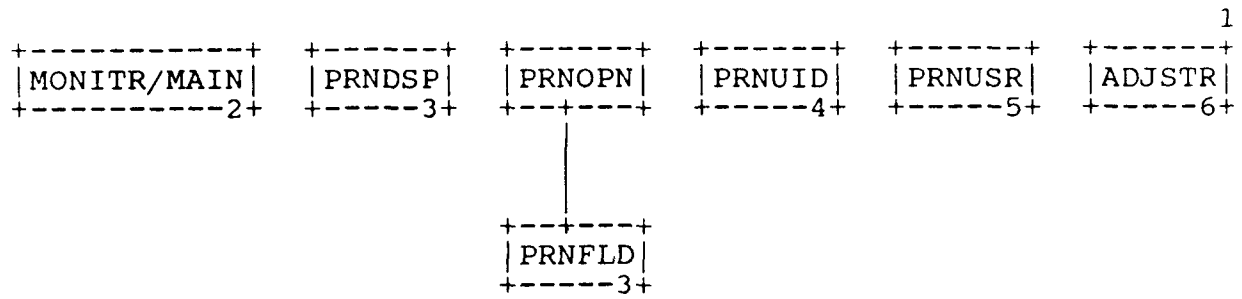
DESCRIPTION:

DESCRIPTION

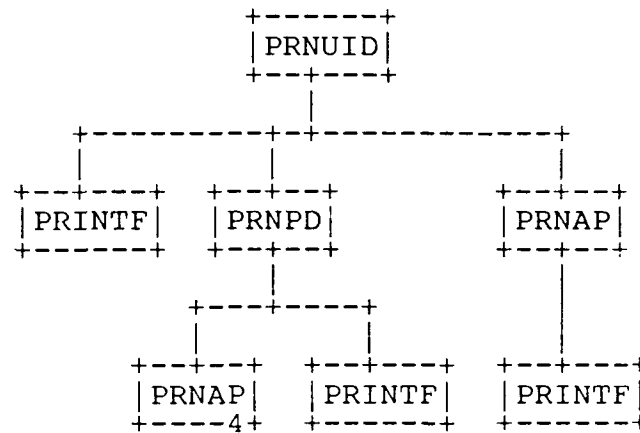
3.10.10 Hierarchy Chart

The following hierarchy charts show the relationships between all of the modules mentioned in the above documentation. A module may call a subroutine several times within its code, but the call will only be shown once as a single relationship on this hierarchy chart. All modules shown at the top of the first page are considered Main Programs as described in section 3.10.1 above.

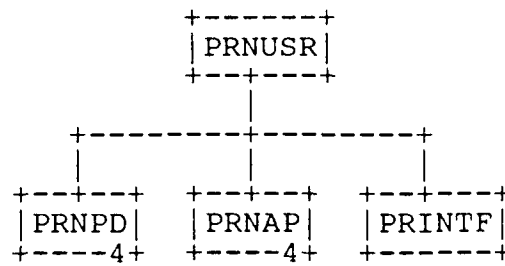
There is an internal paging scheme as marked by the numbers in the upper right corner of each page. An index after the last page of the chart shows where a routine and its calls are first defined. If a routine has no page reference, it either makes no calls or is an external routine. A continuation box on the end of a tree limb shows where that the tree continues on the page numbered mentioned. A number in a box with a routine name points to the page where the routine is further defined within the hierarchy tree. If there is no number in a box, the routine either makes no calls or is an external routine.



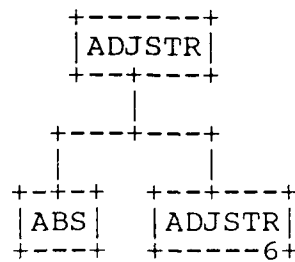
4



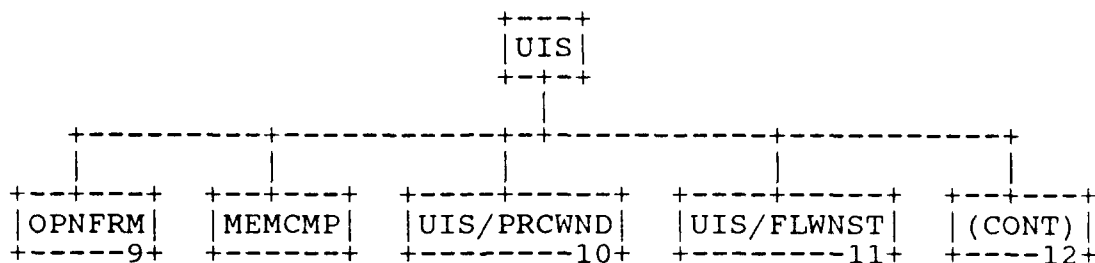
5



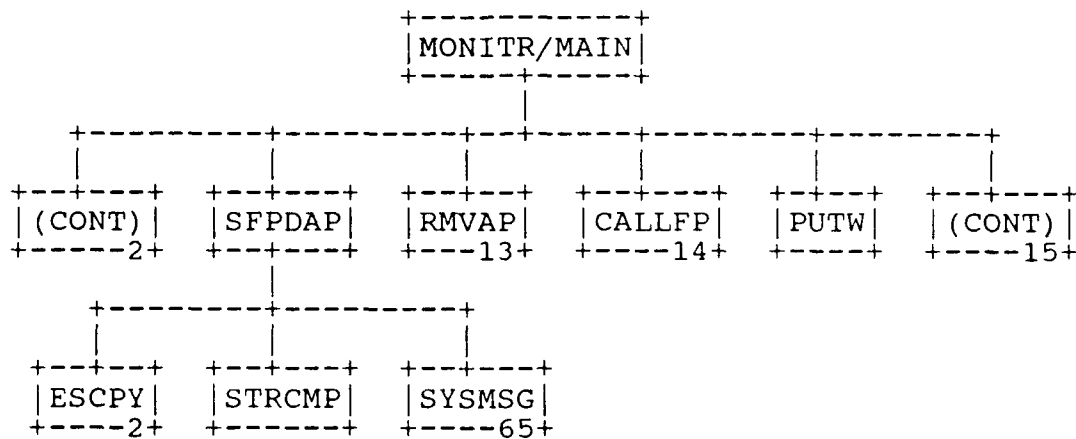
6



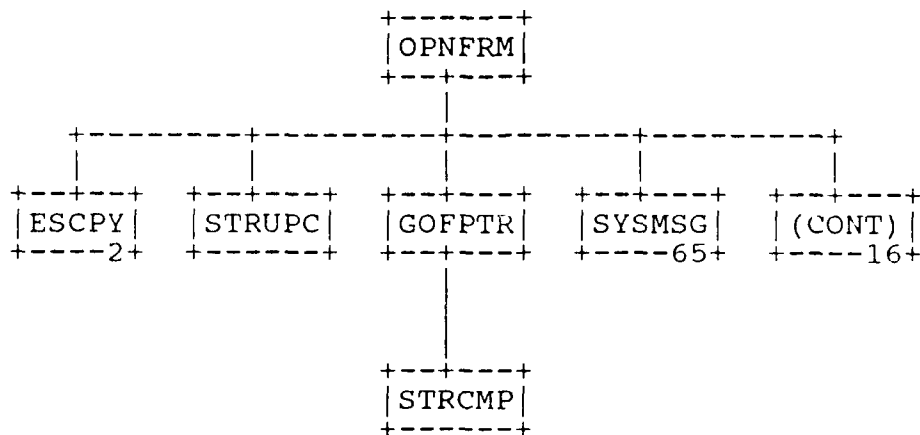
7



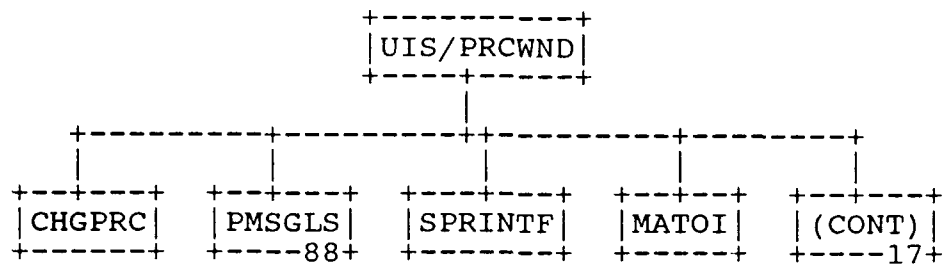
8



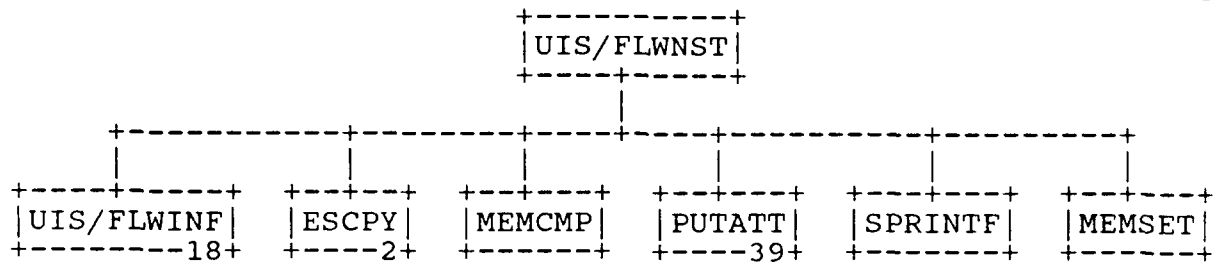
9



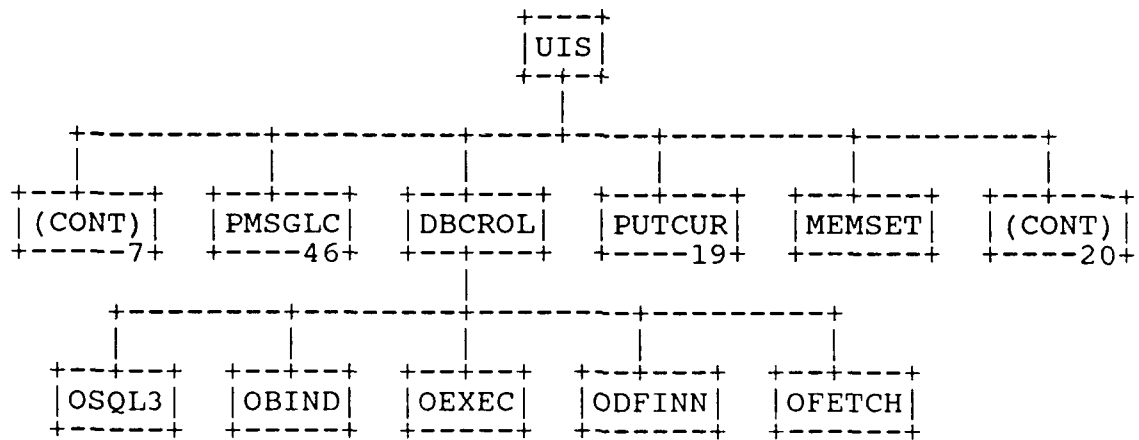
10



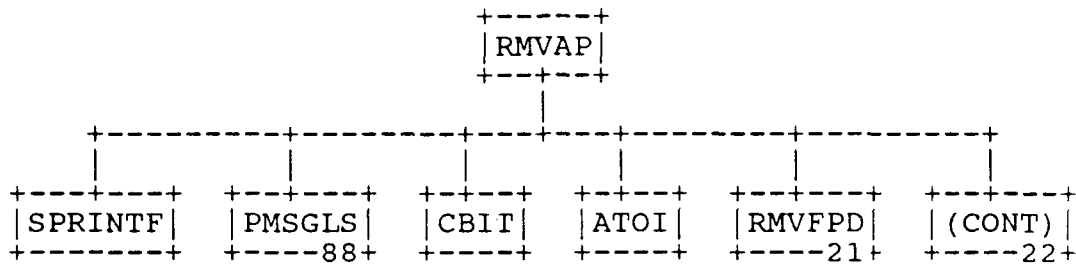
11



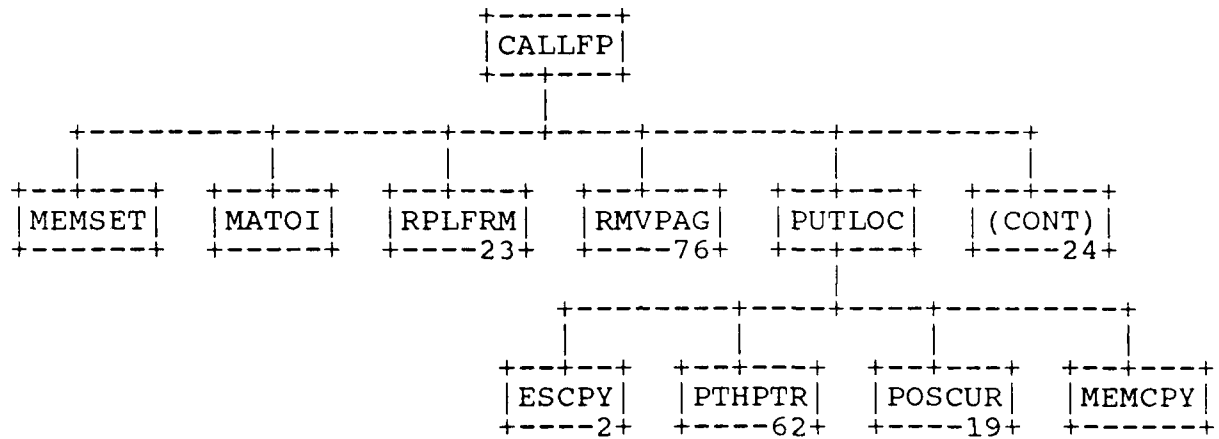
12



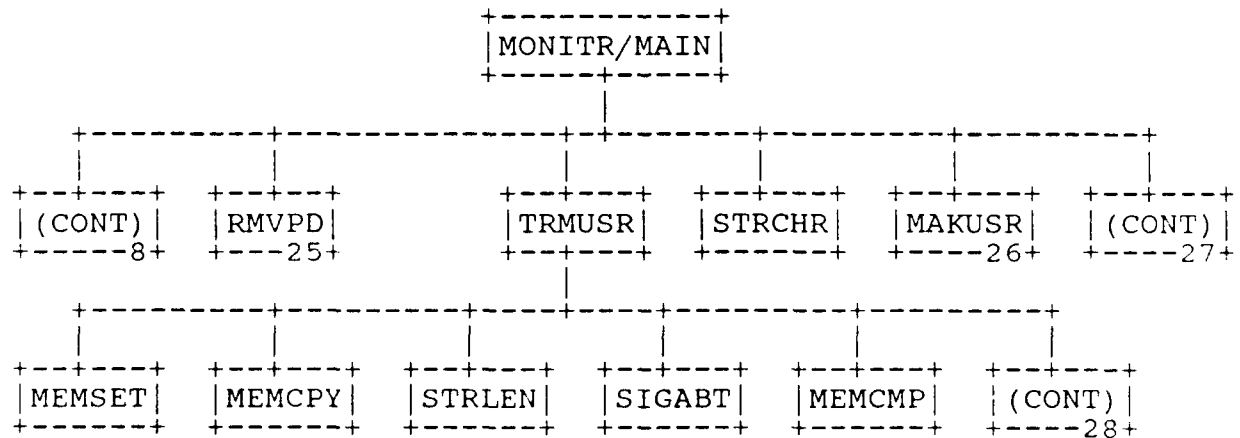
13



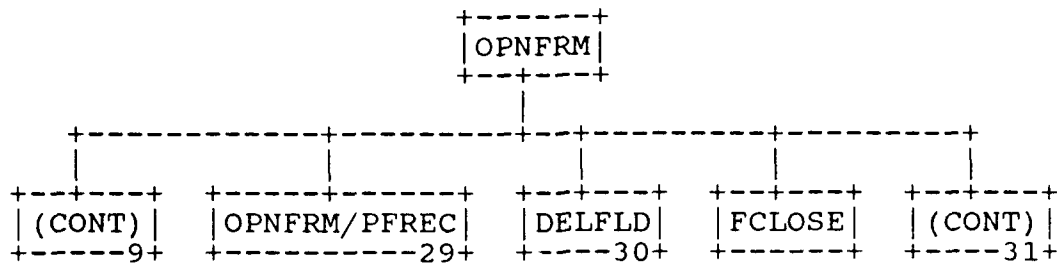
14



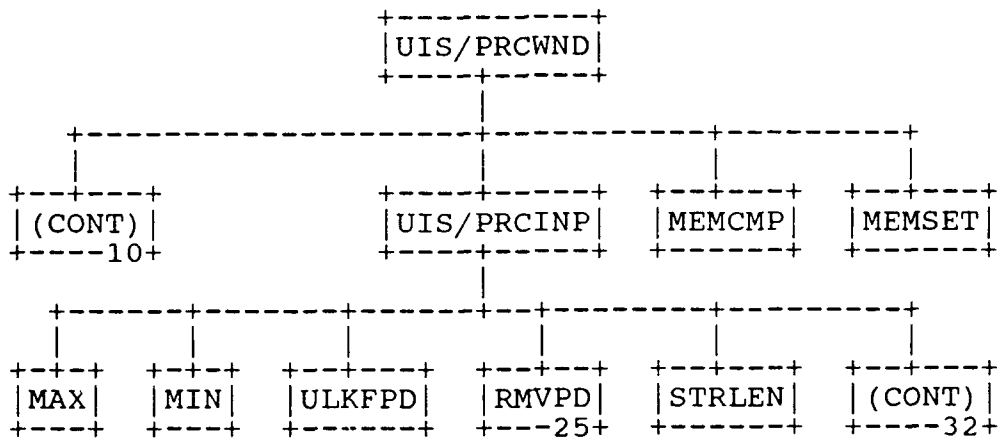
15



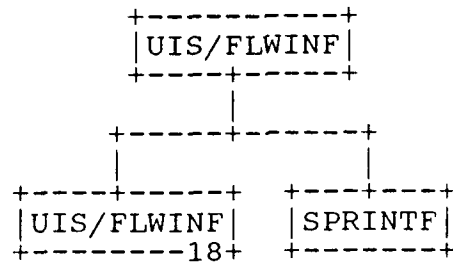
16

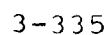


17

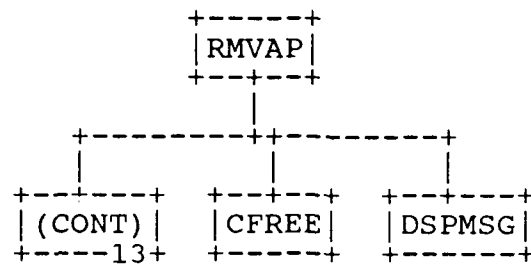


18

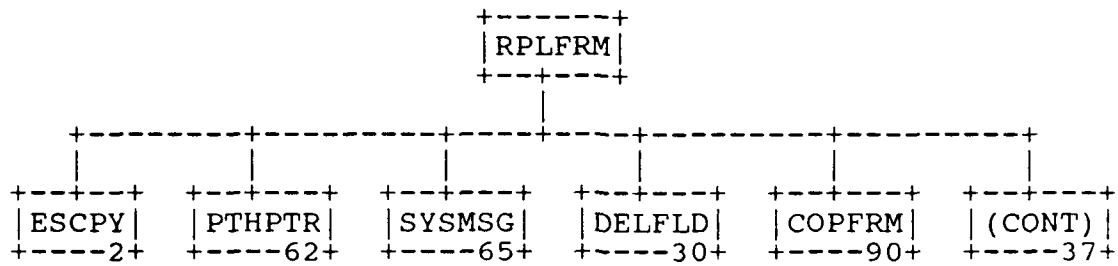




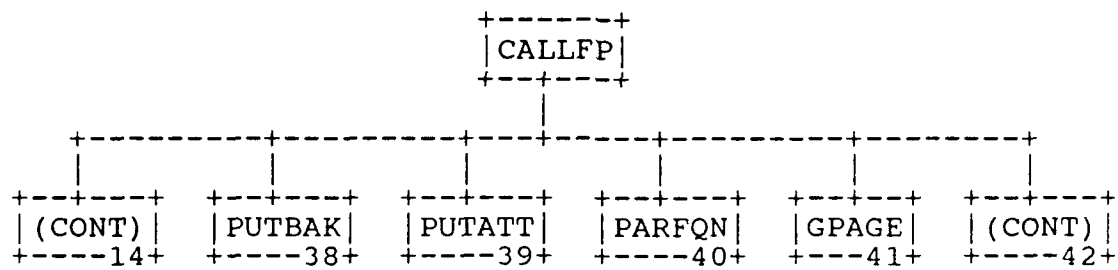
22



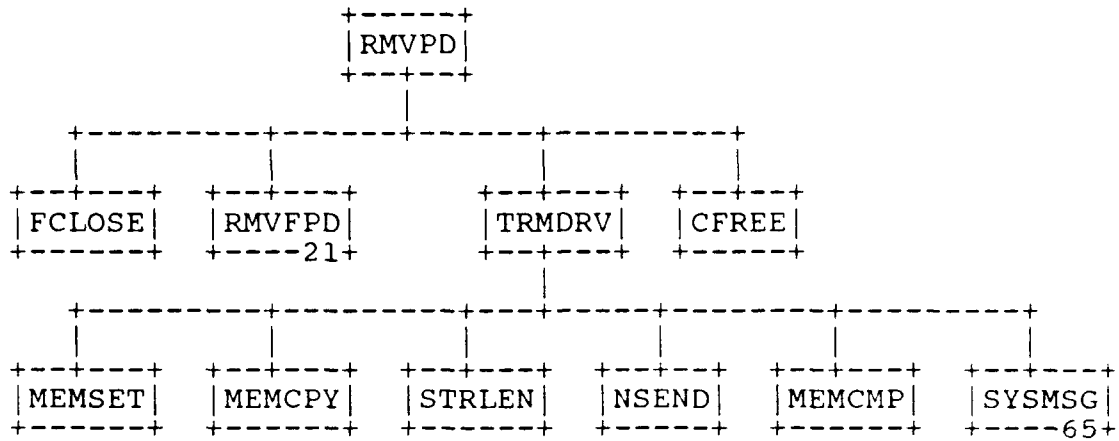
23



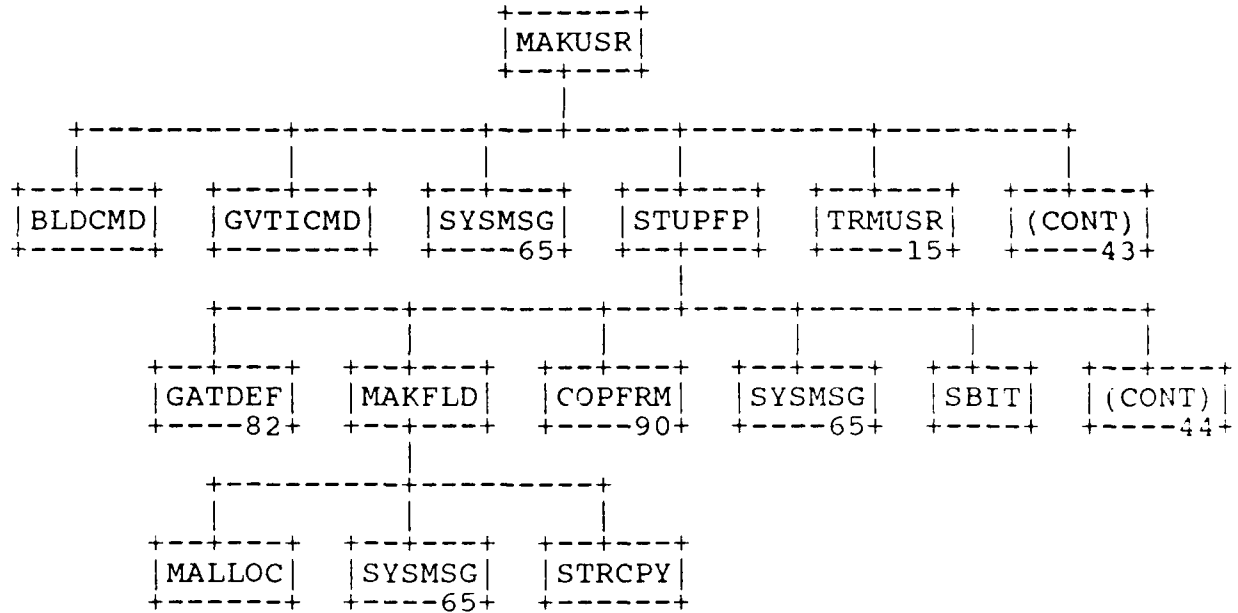
24



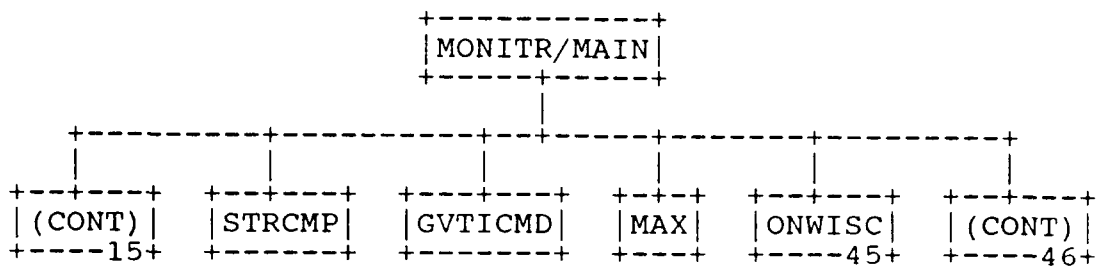
25



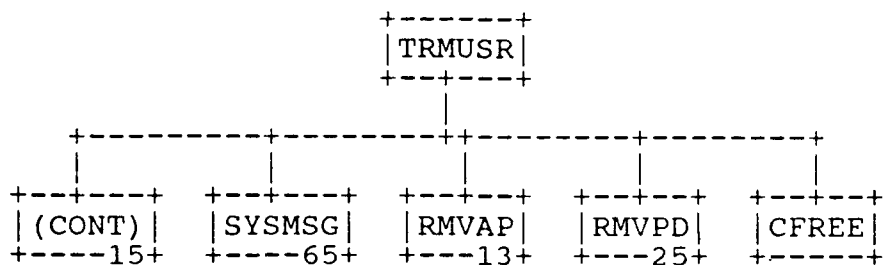
26



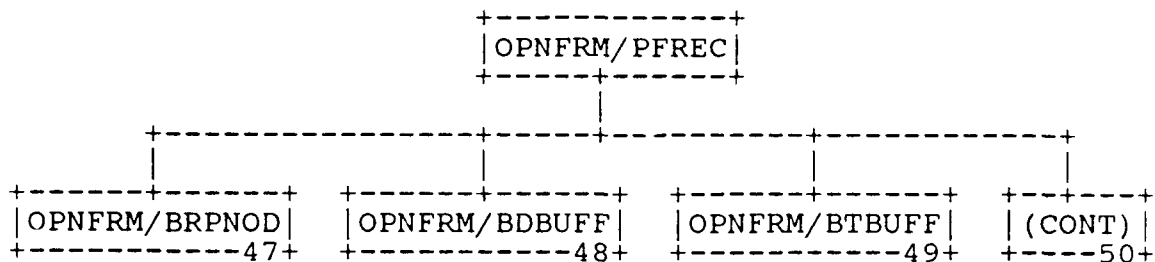
27



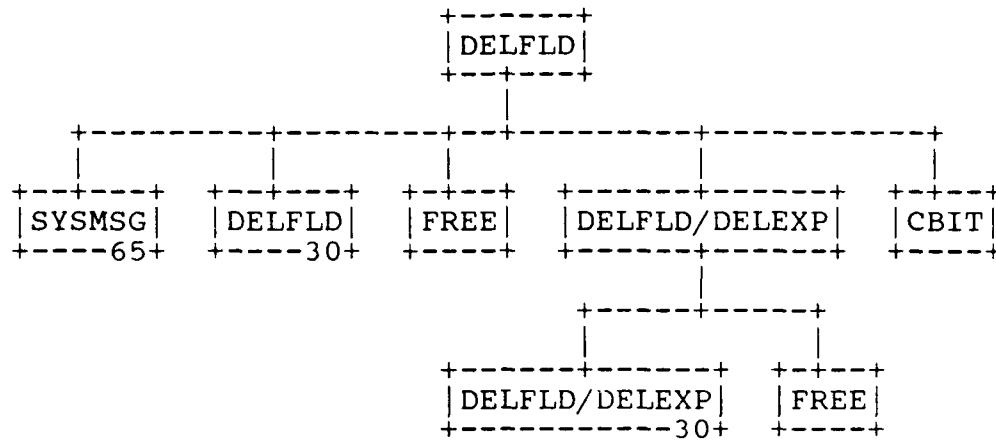
28



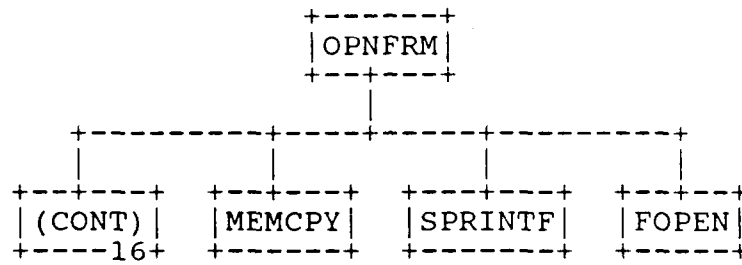
29



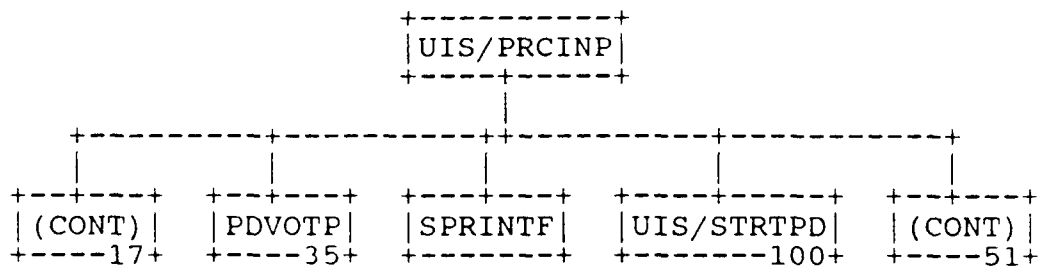
30



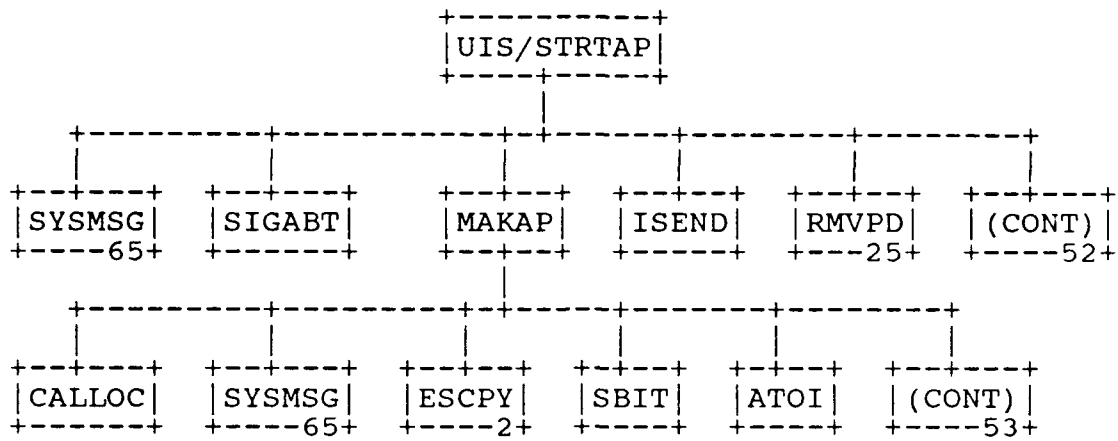
31



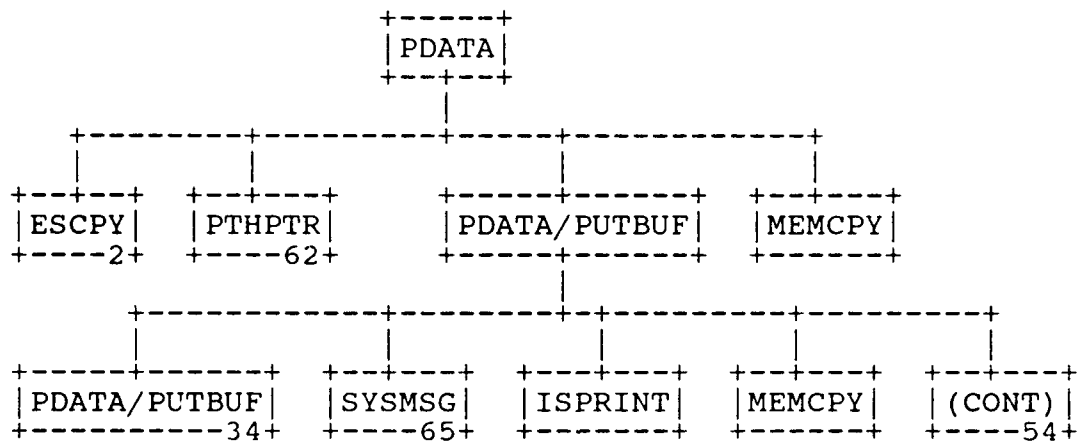
32



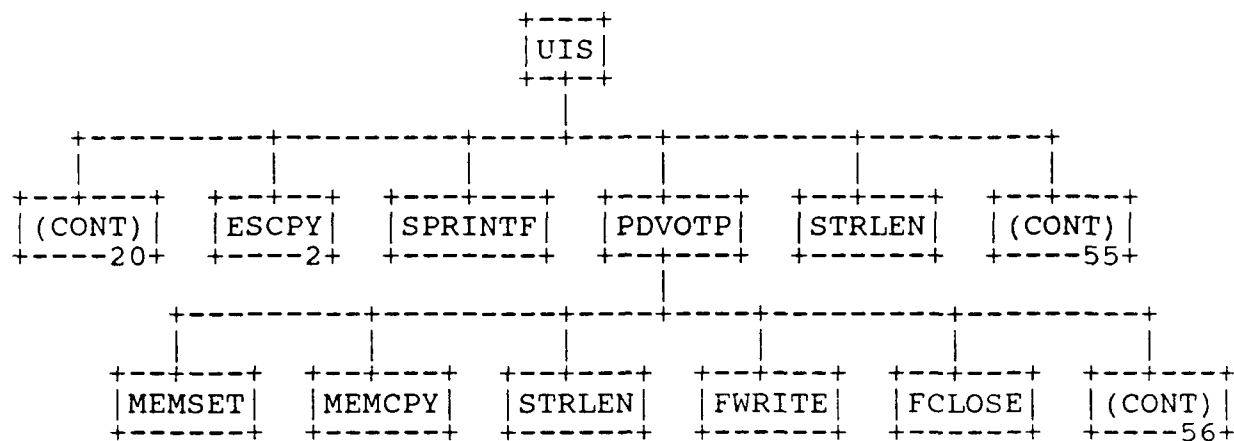
33



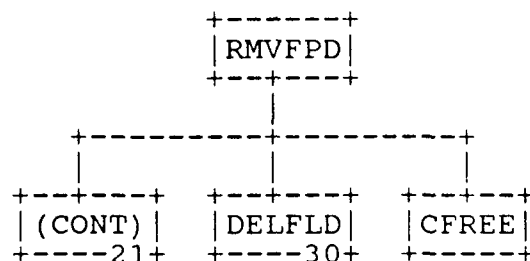
34



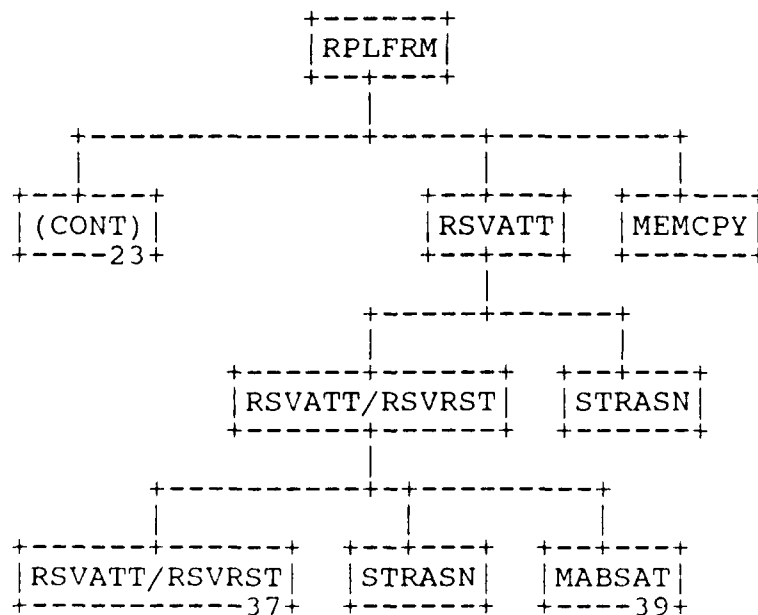
35



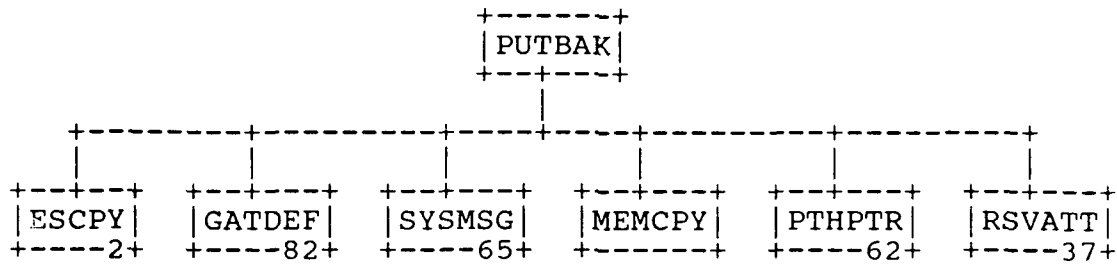
36



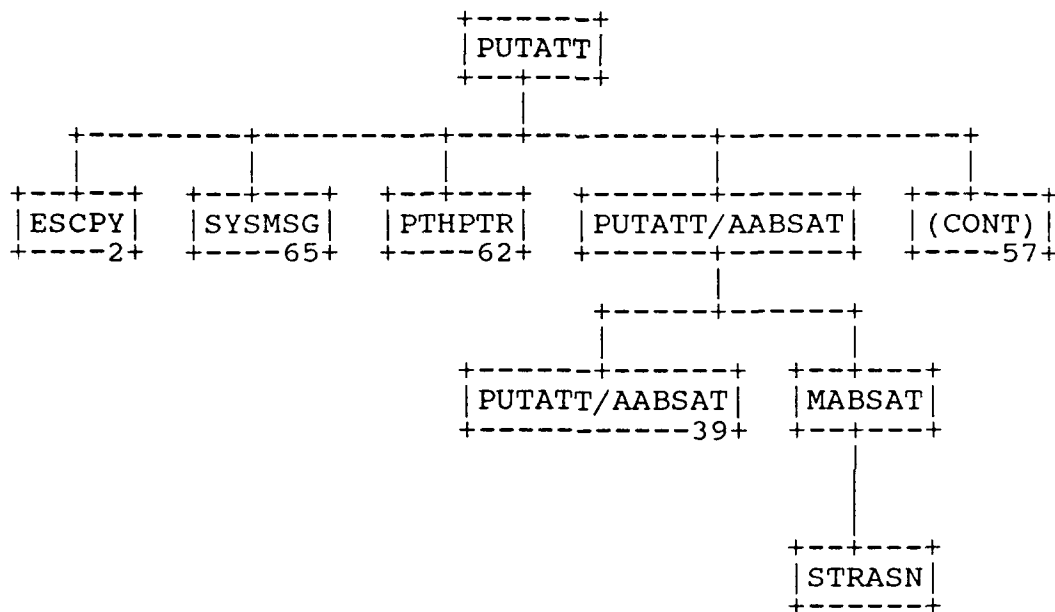
37



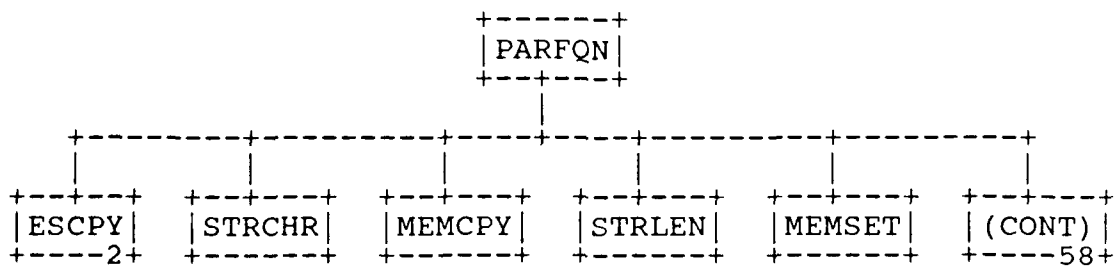
38



39

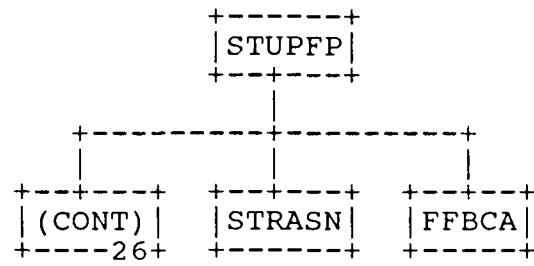


40

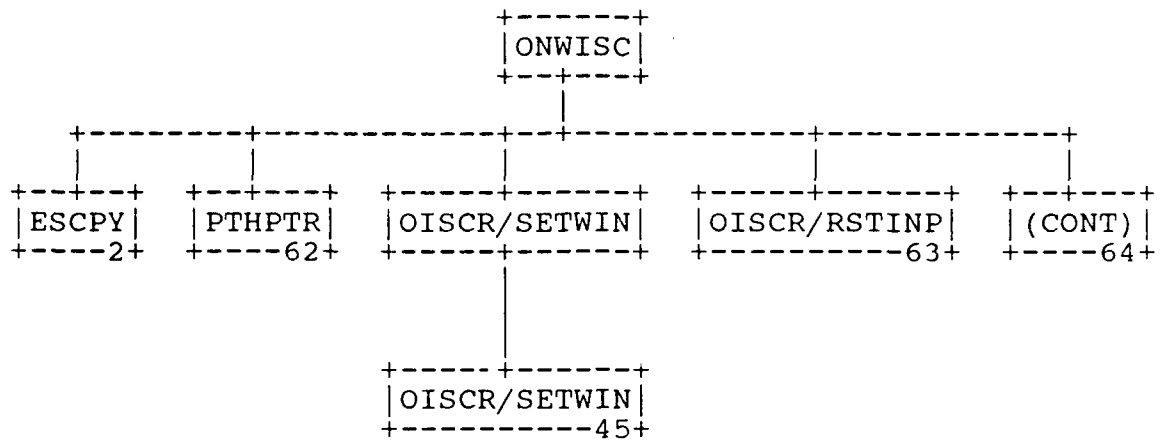




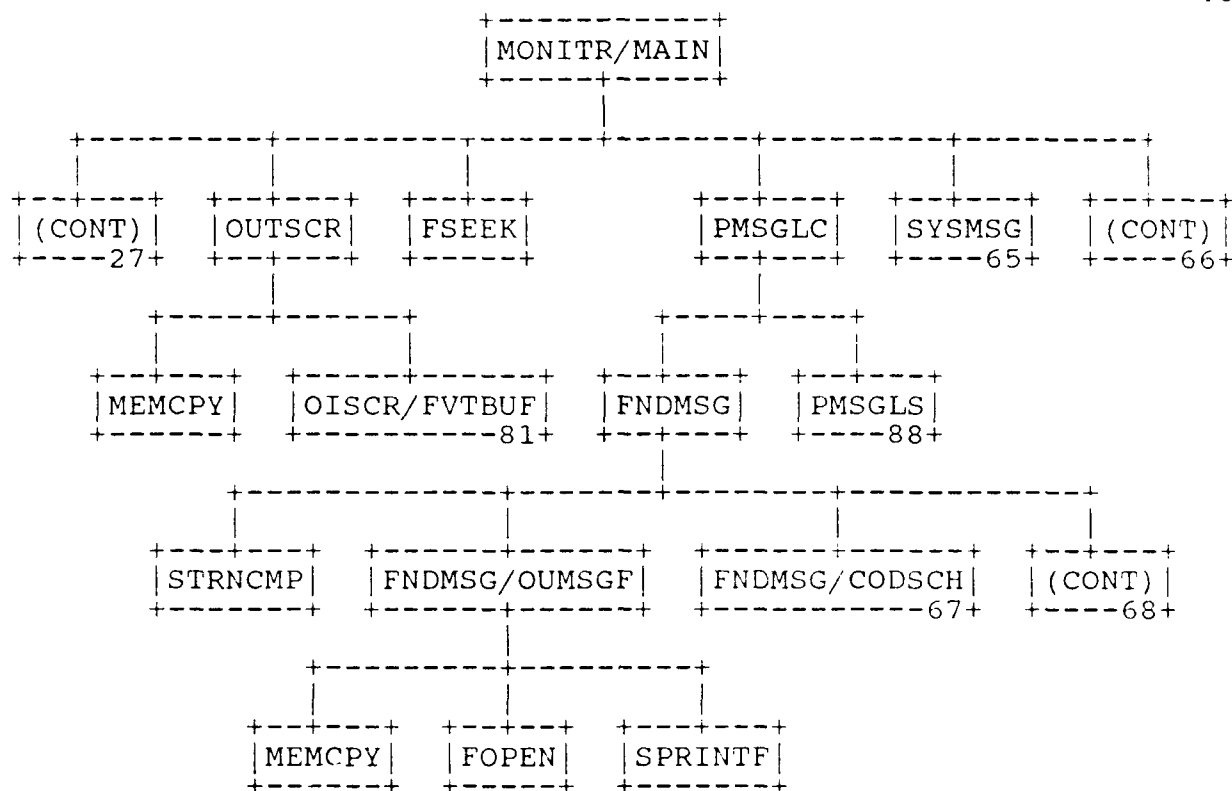
44



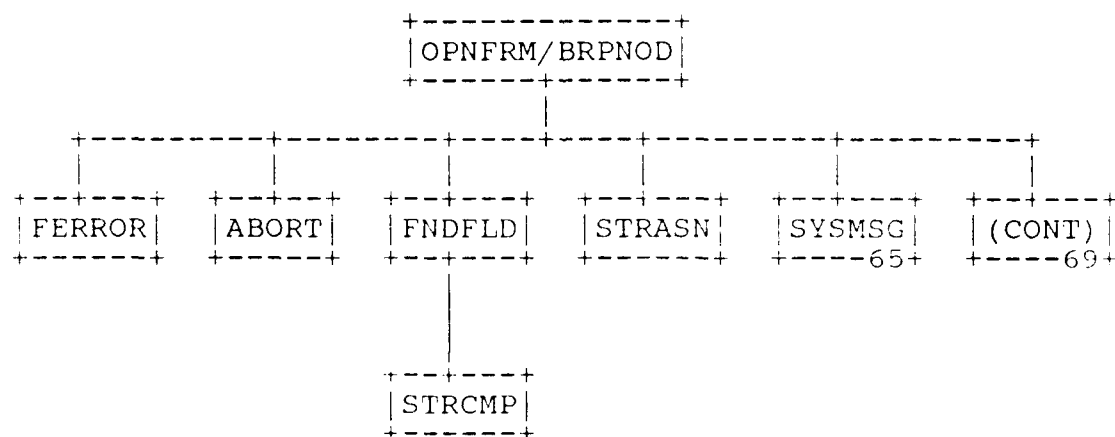
45



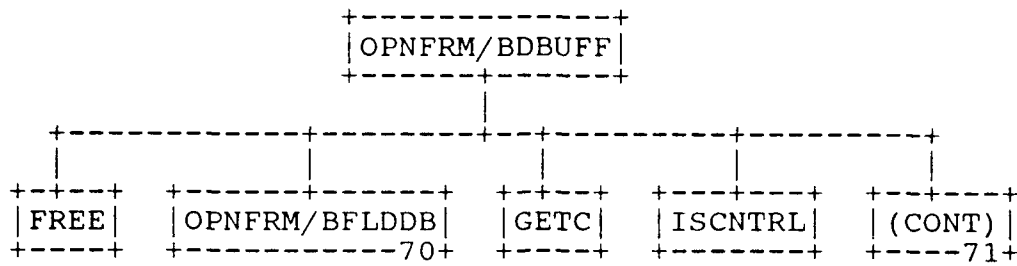
46



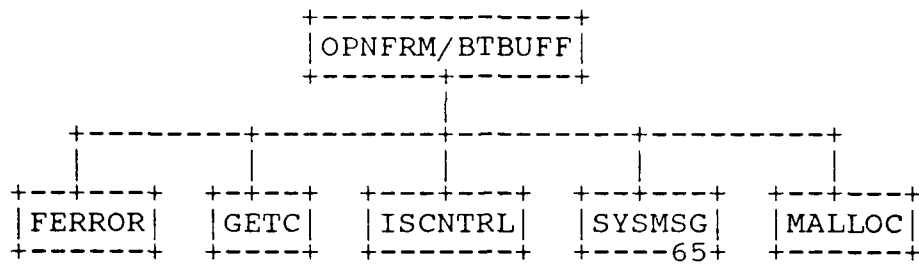
47



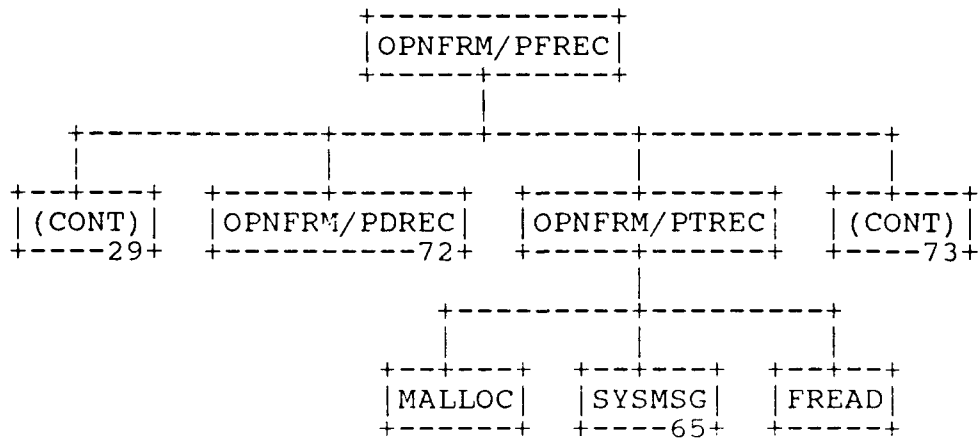
48



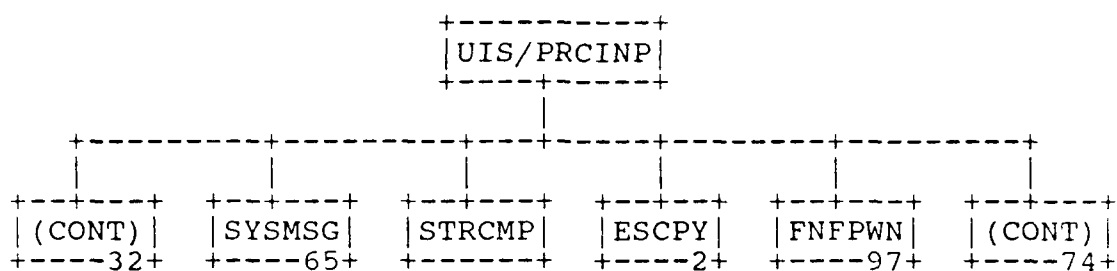
49



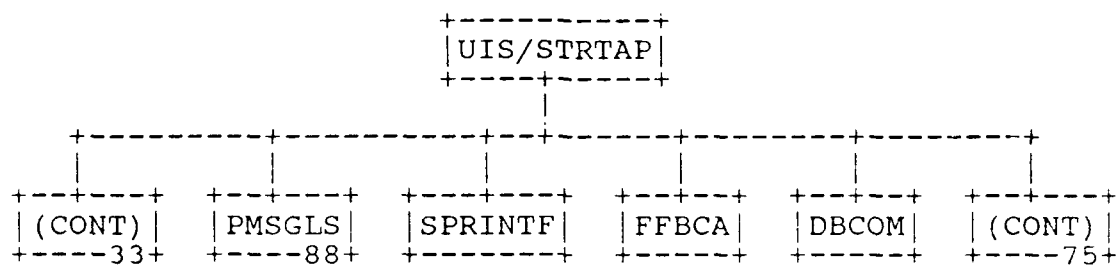
50



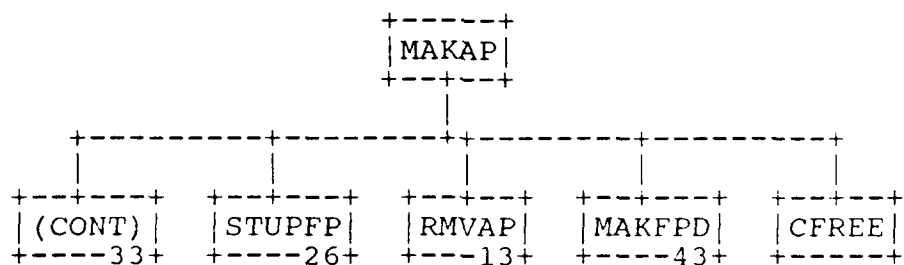
51



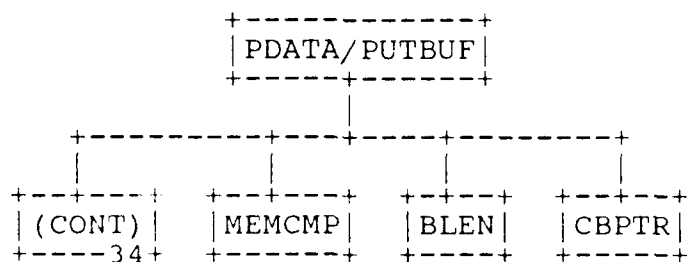
52



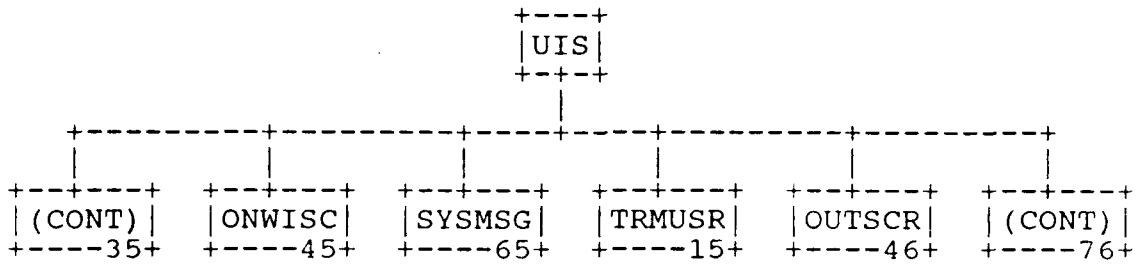
53



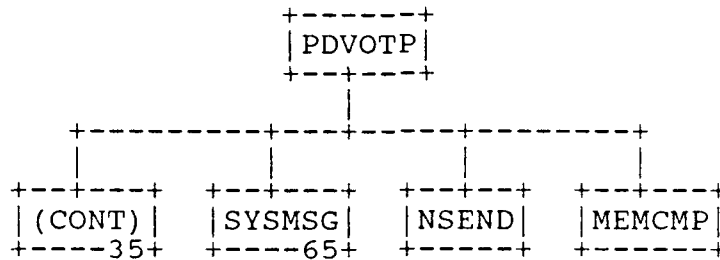
54



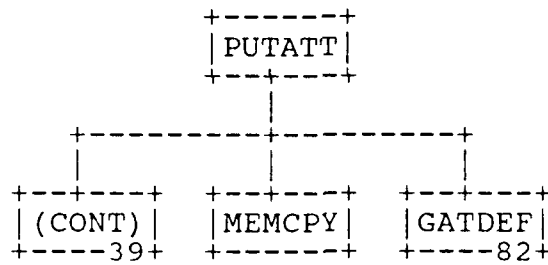
55



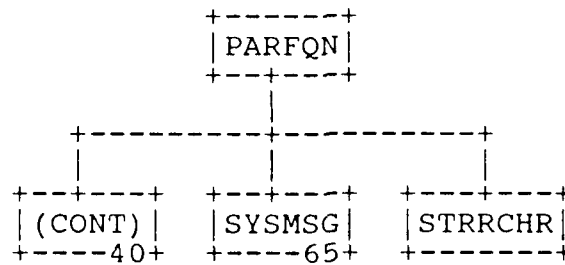
56



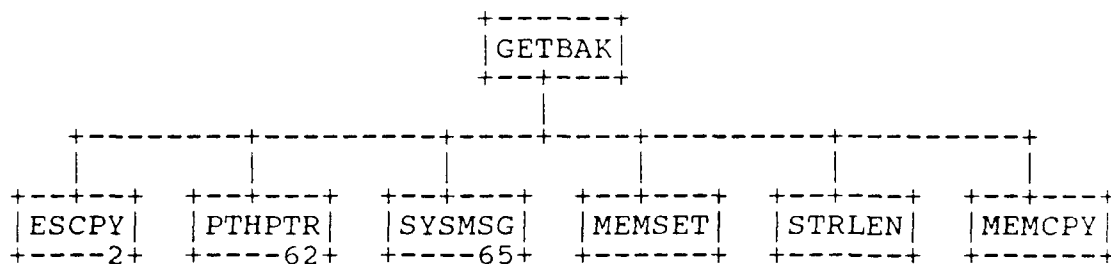
57



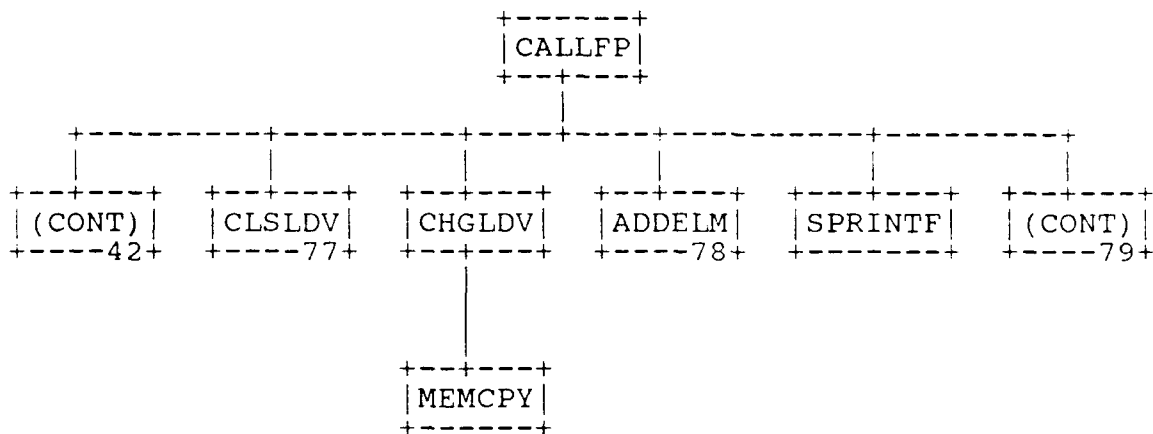
58



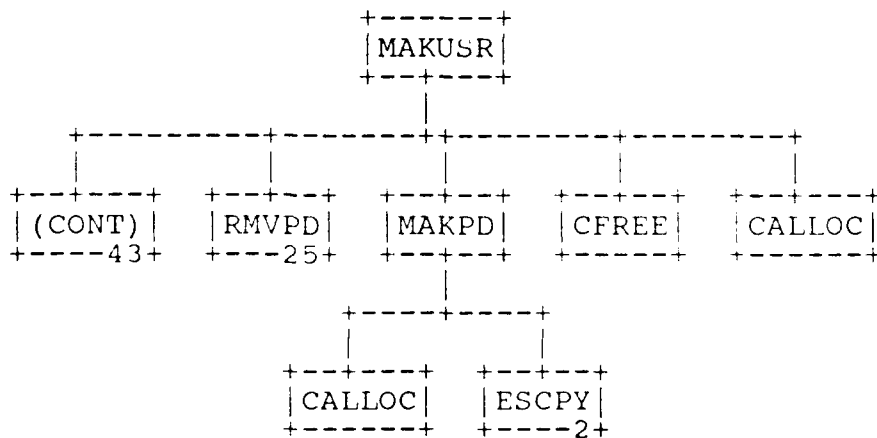
59



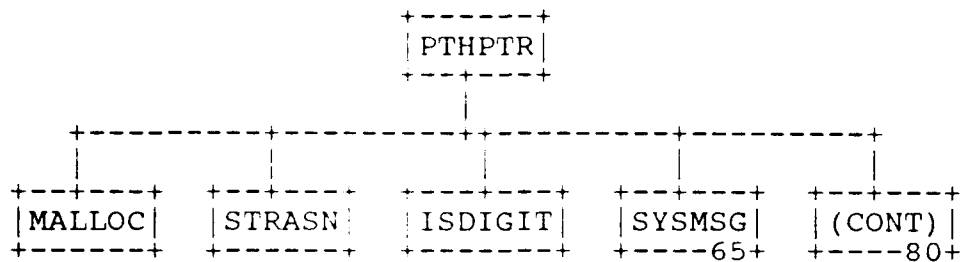
60



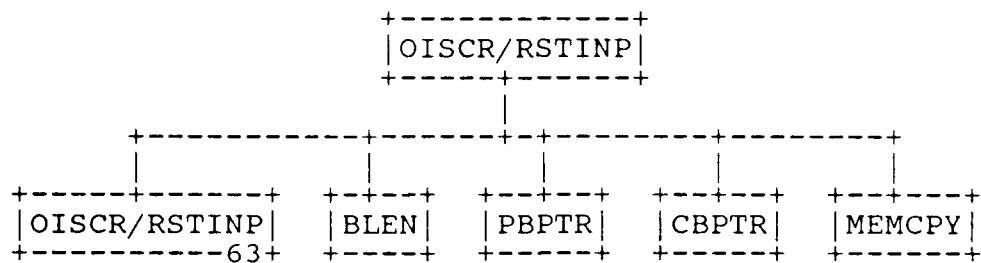
61



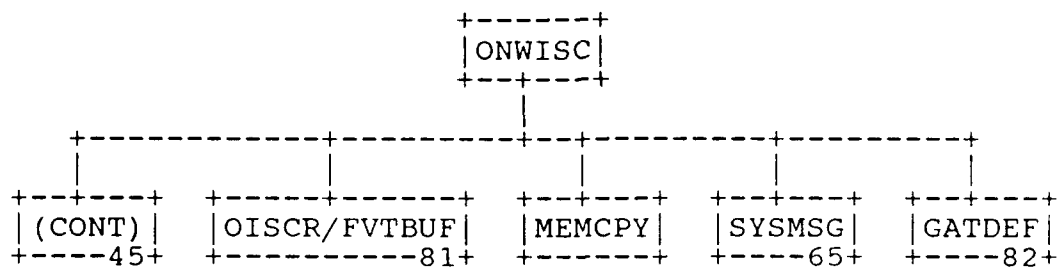
62



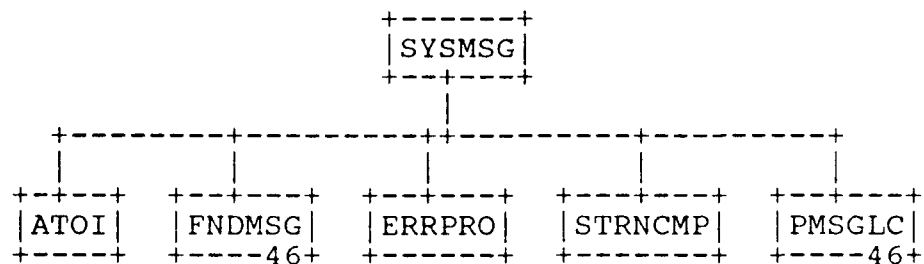
63



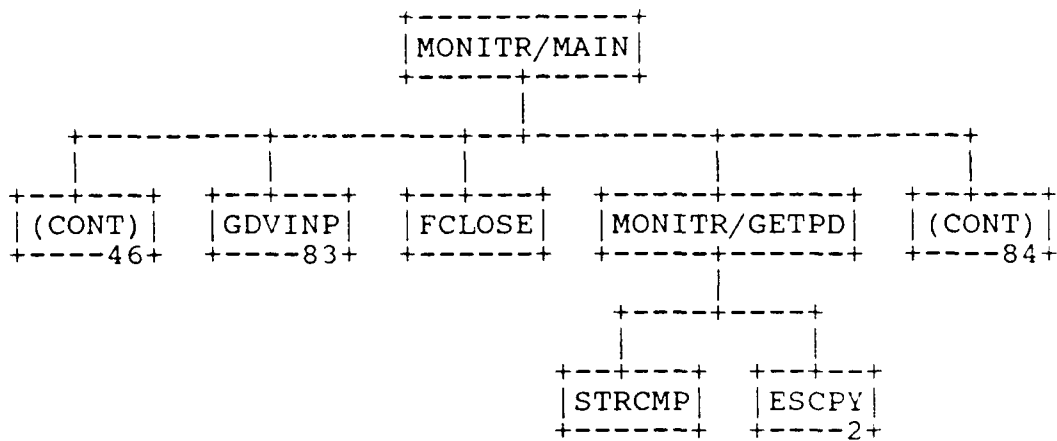
64



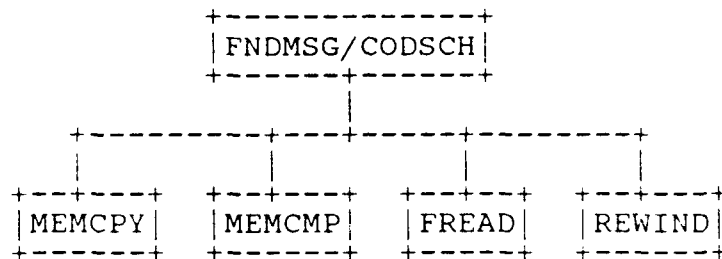
65



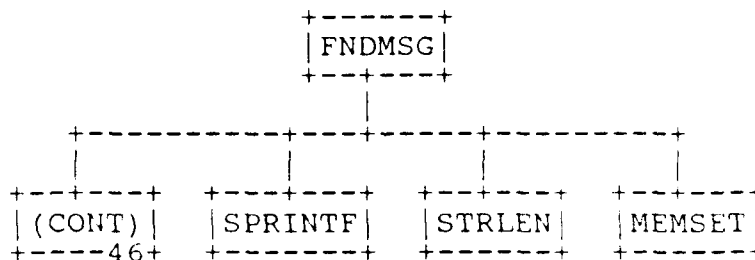
66



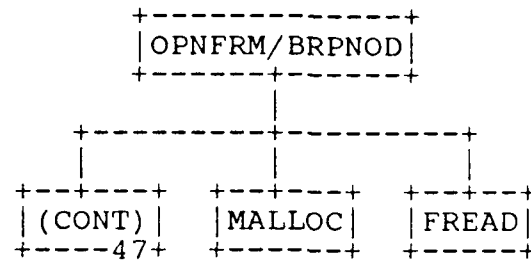
67



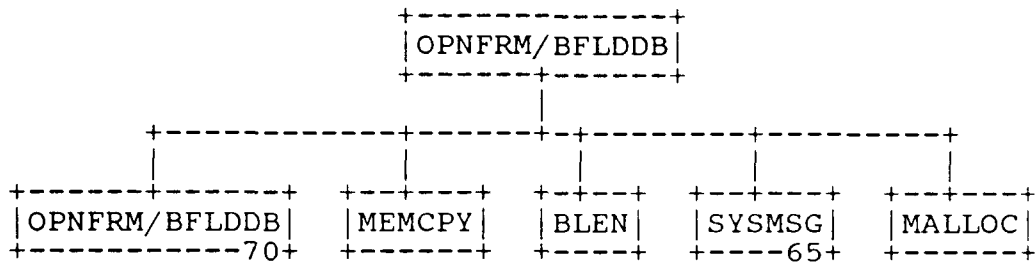
68



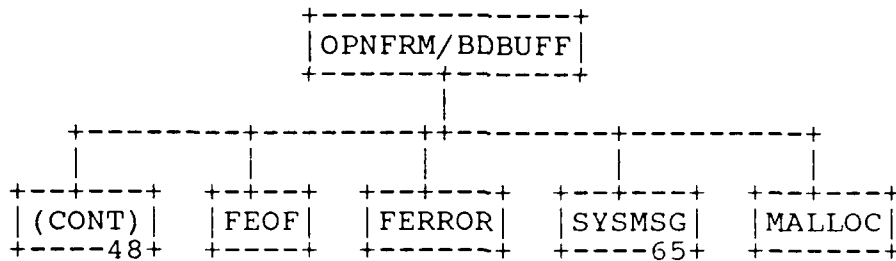
69



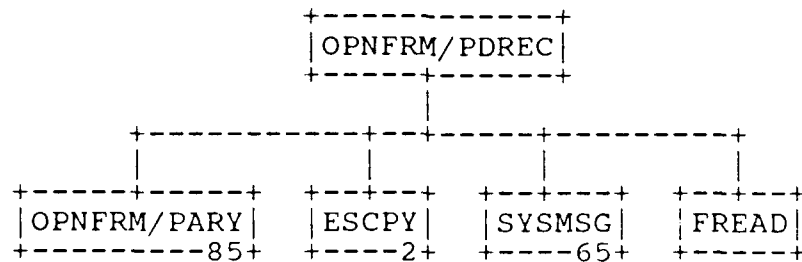
70



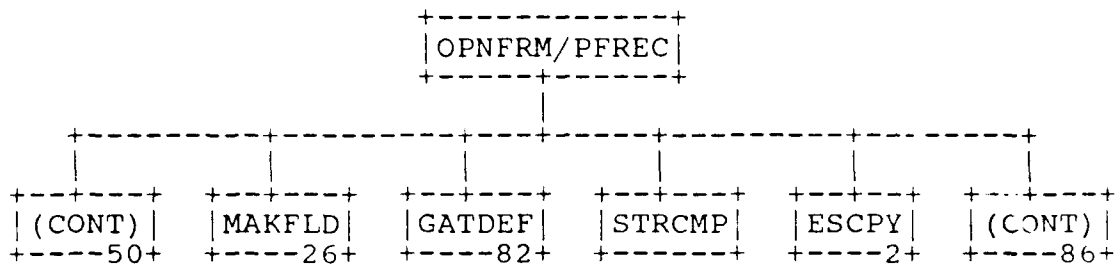
71



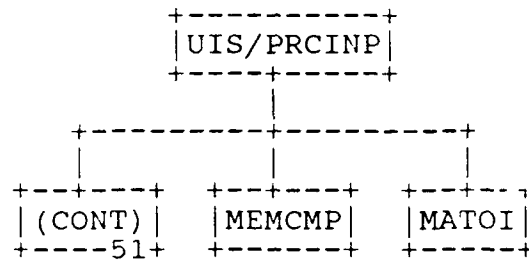
72



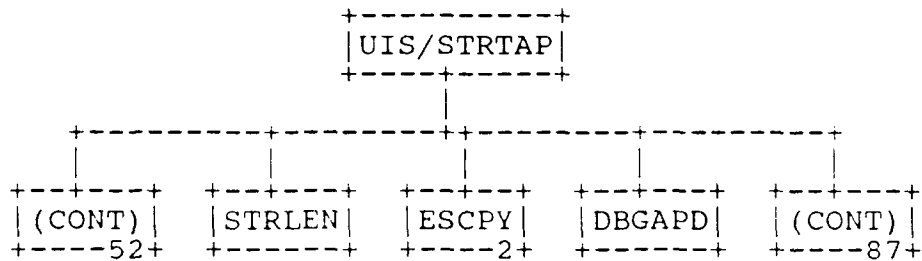
73



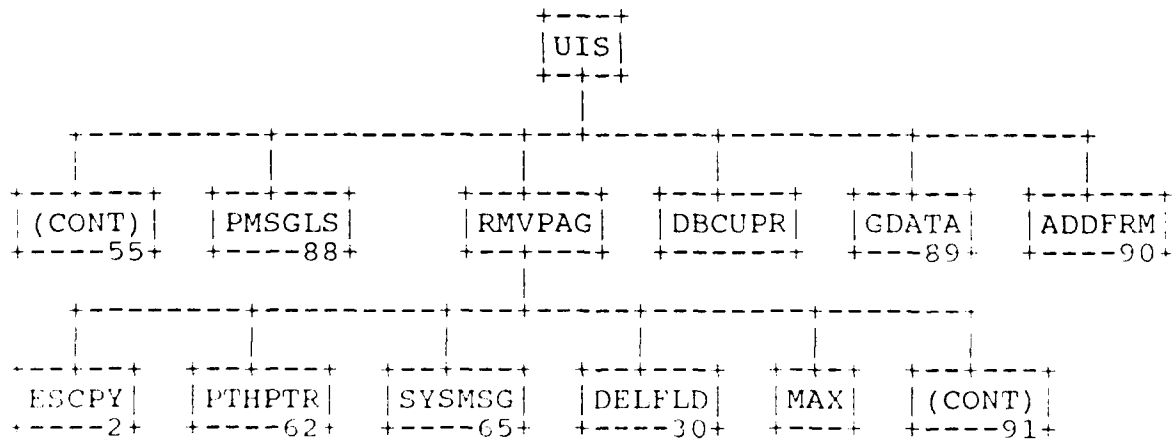
74



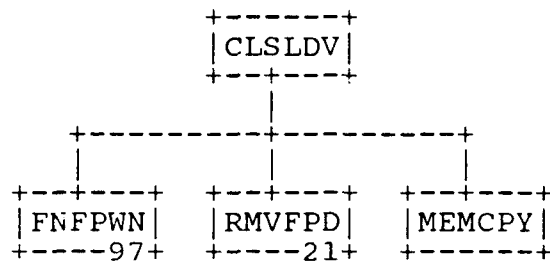
75



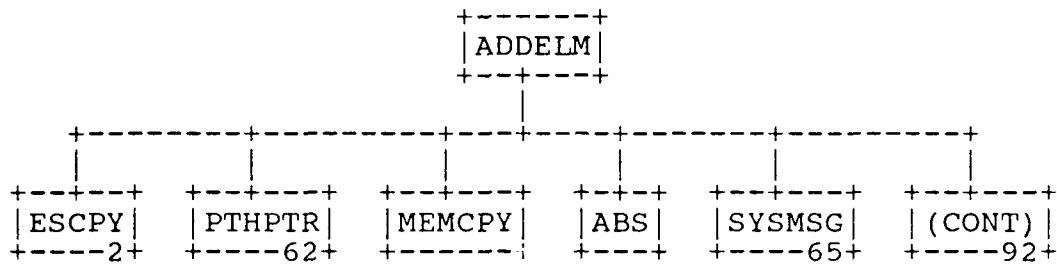
76



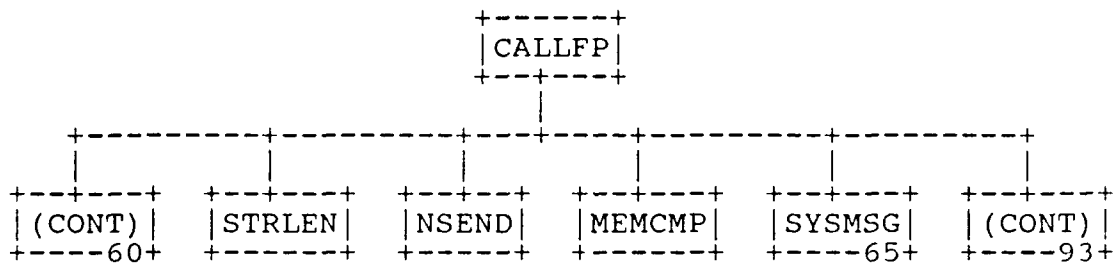
77



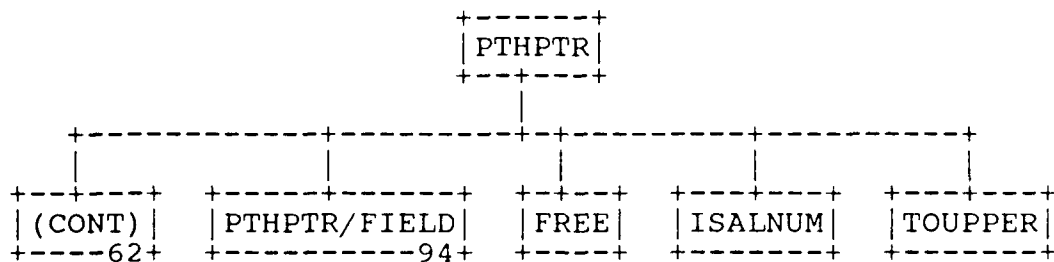
78

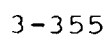


79

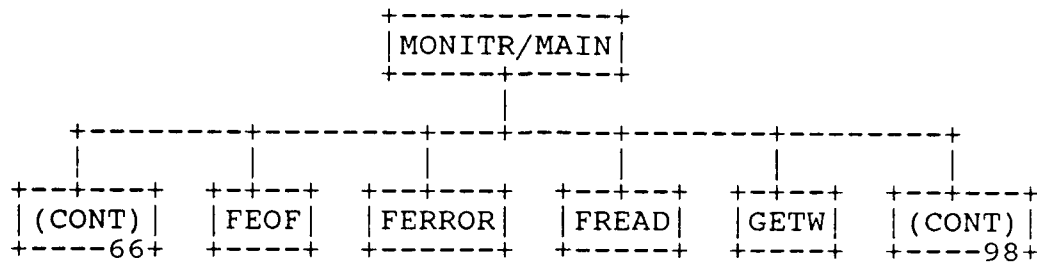


80

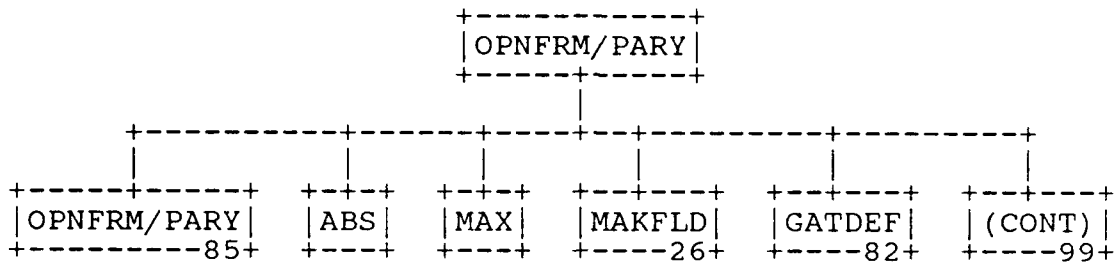




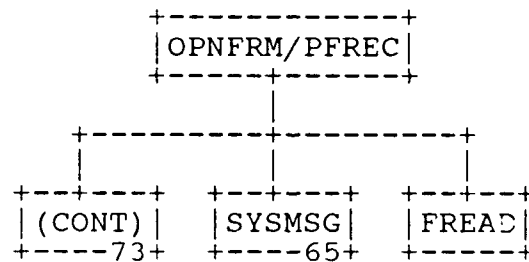
84



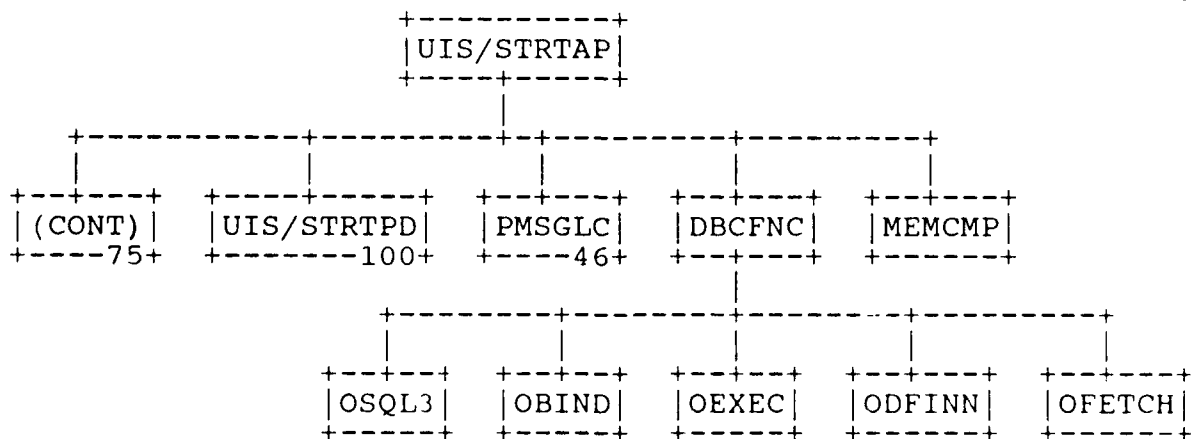
85



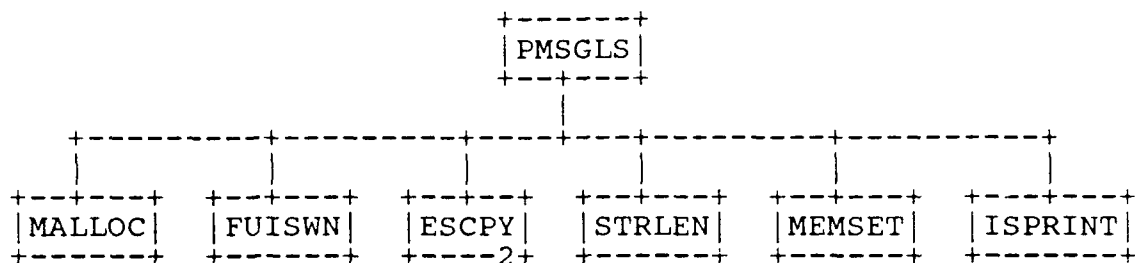
86



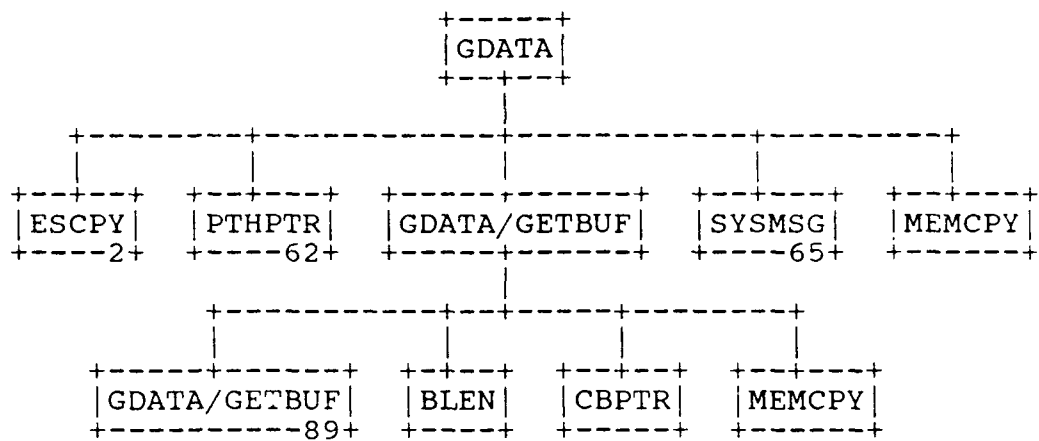
87



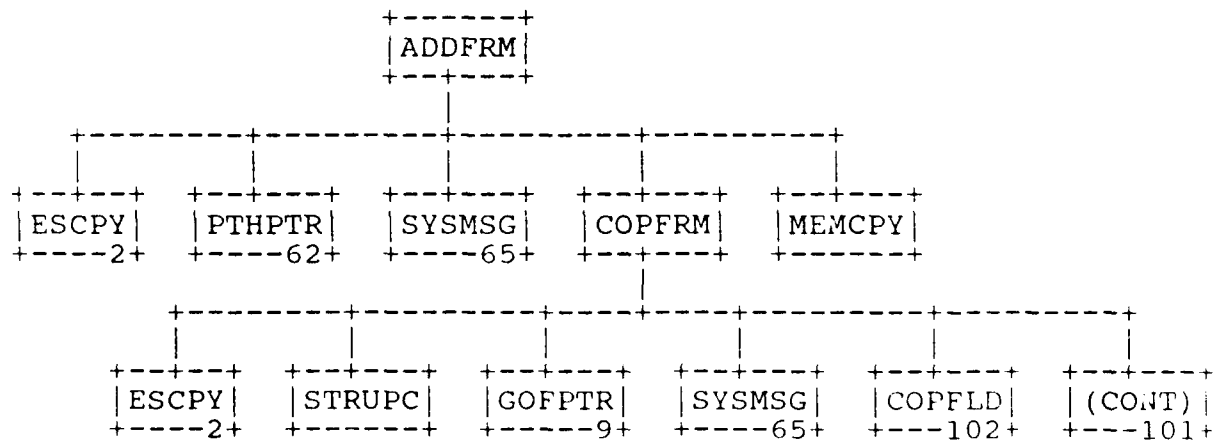
88



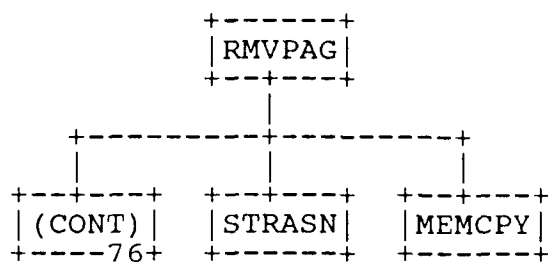
89



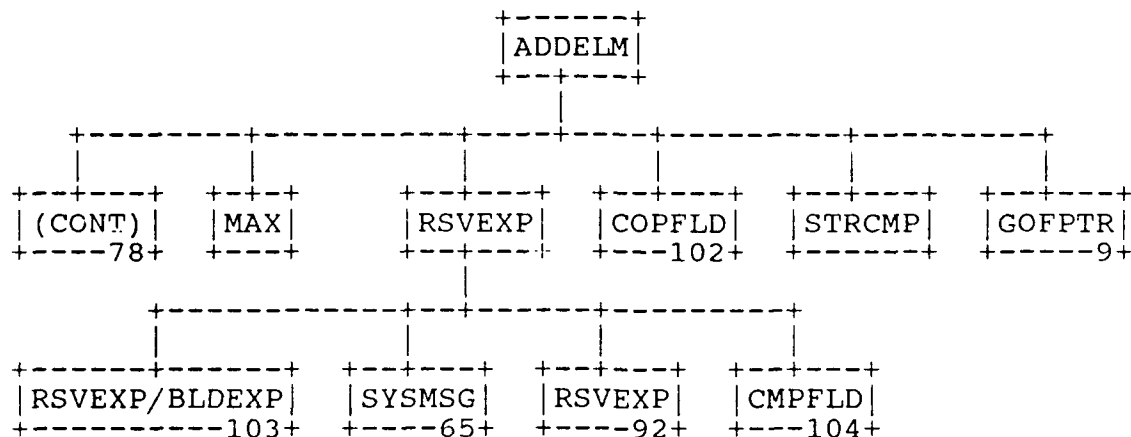
90



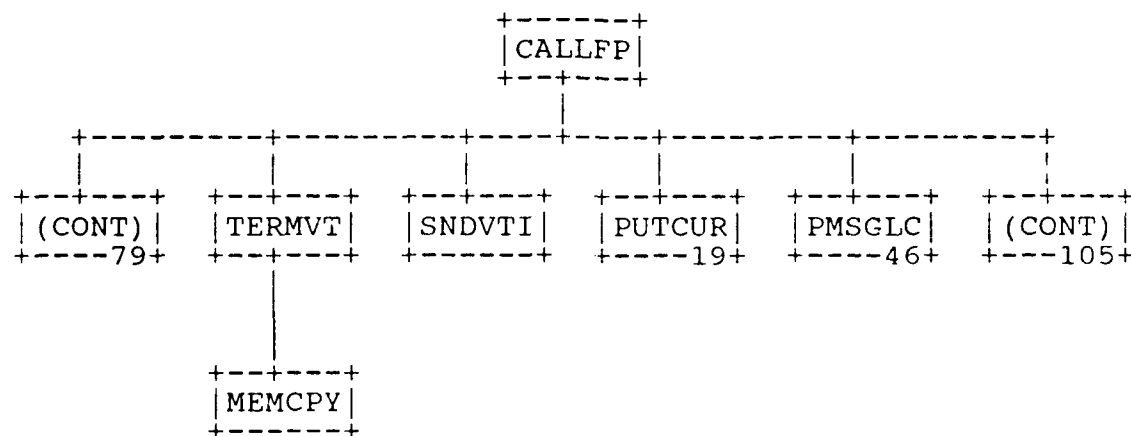
91



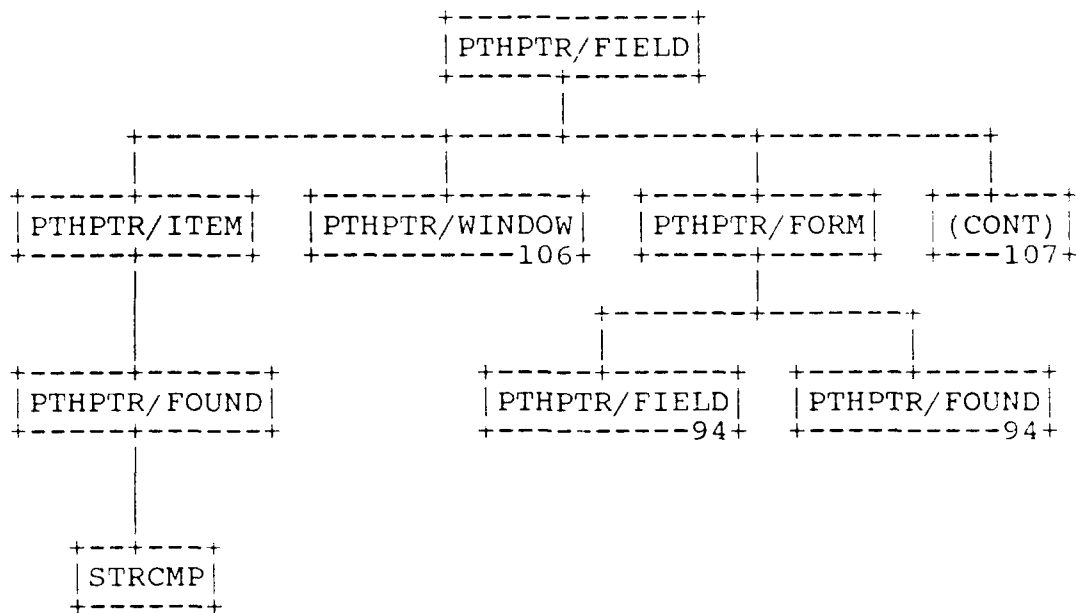
92



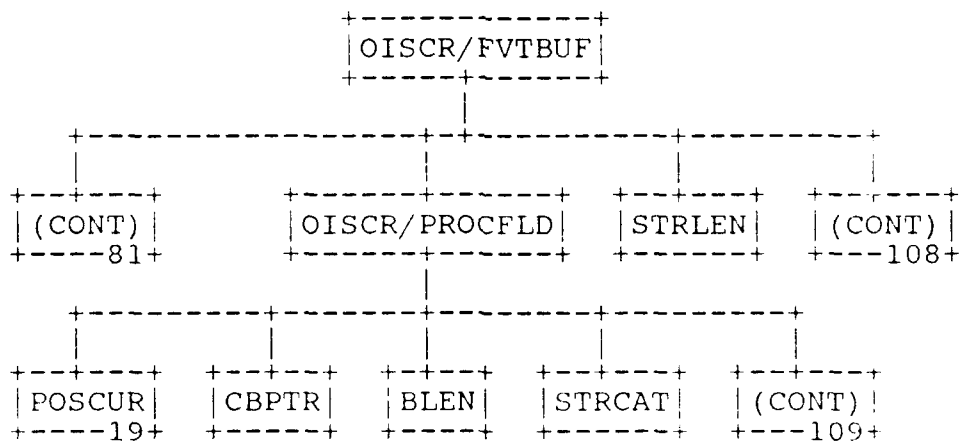
93



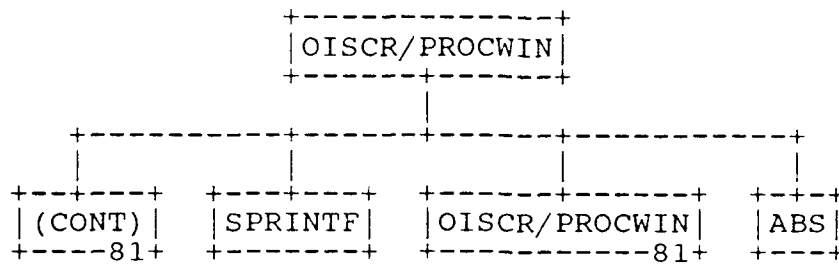
94



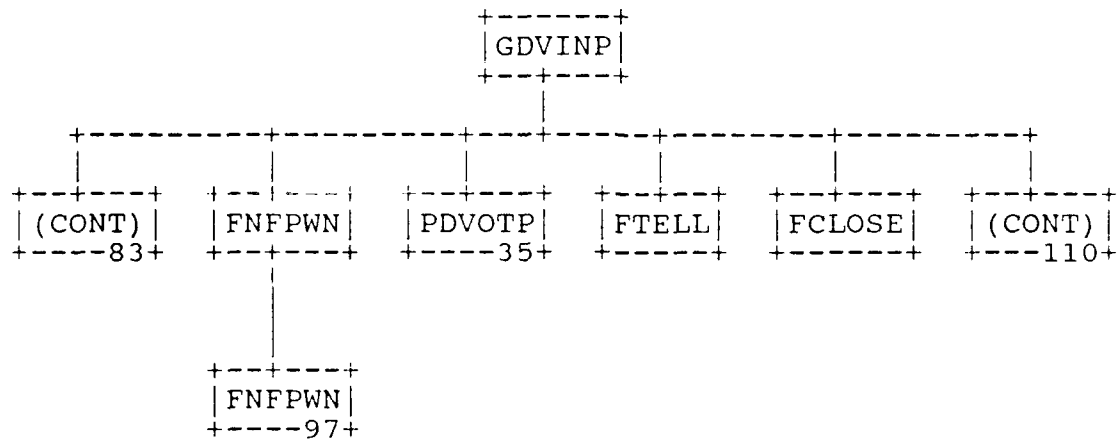
95



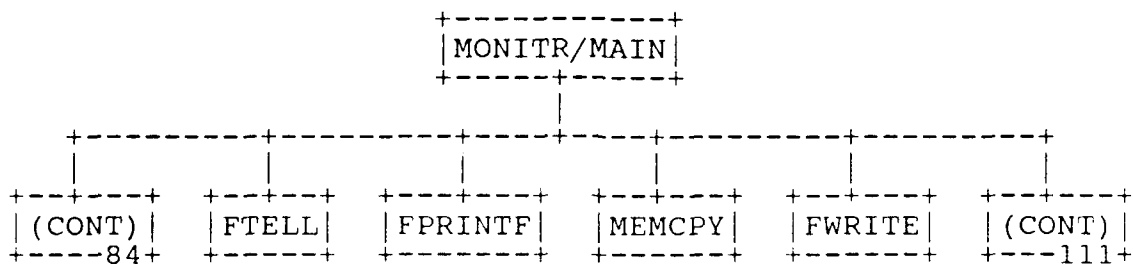
96



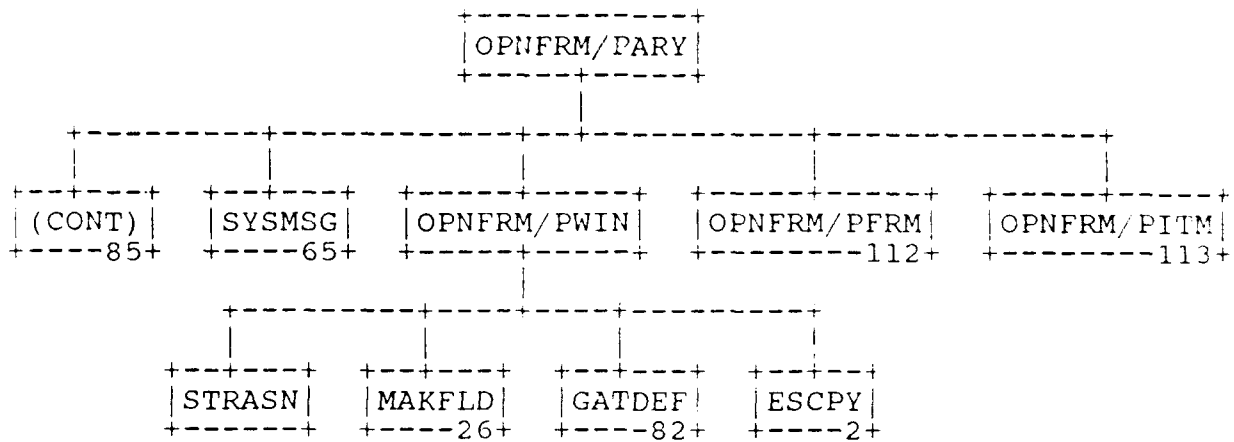
97



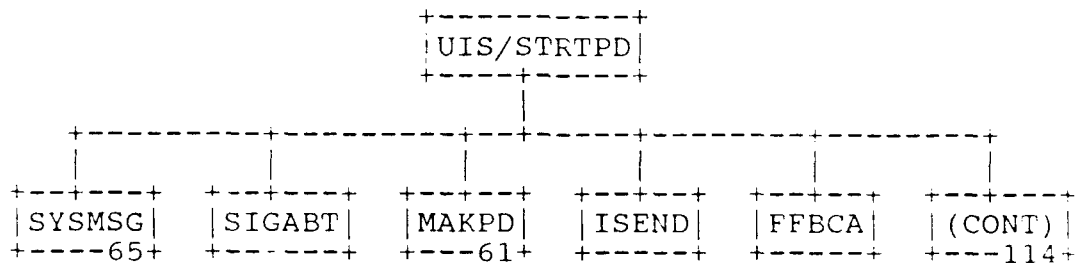
98



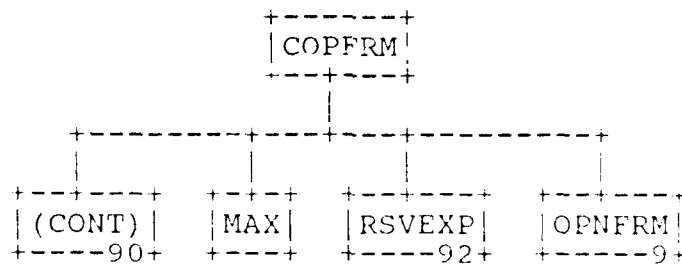
99



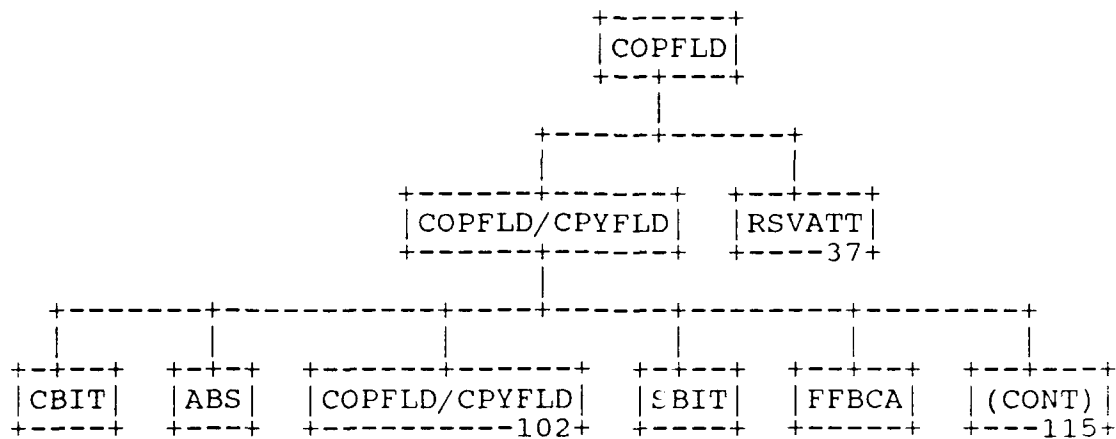
100



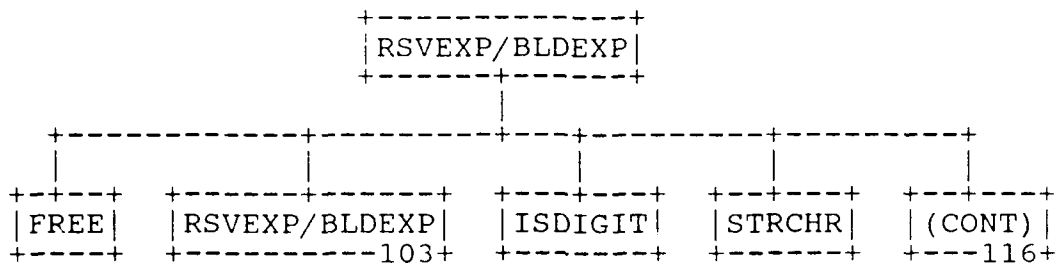
101



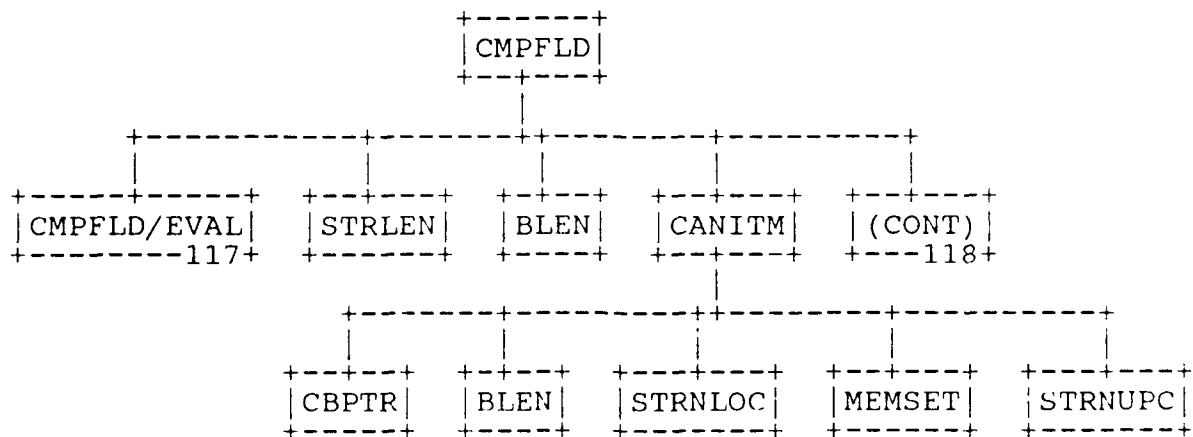
102



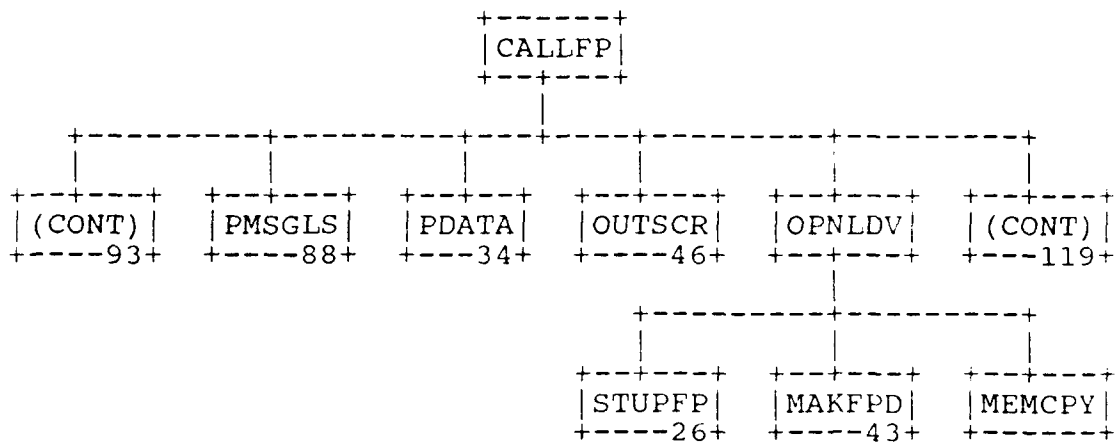
103



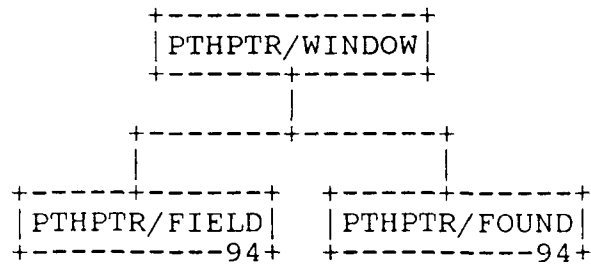
104



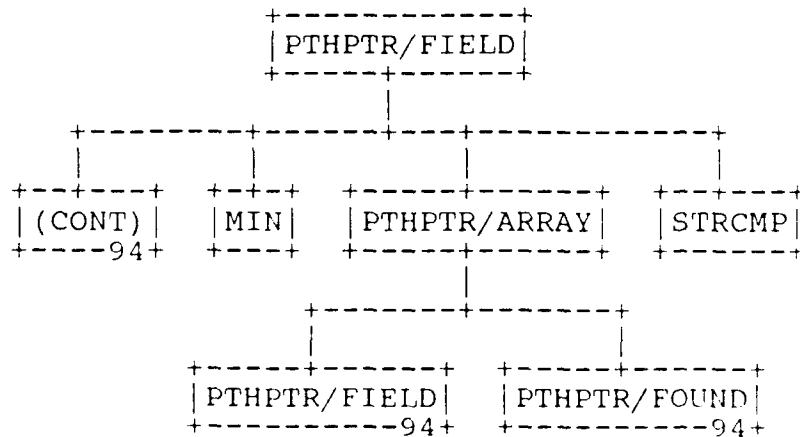
105



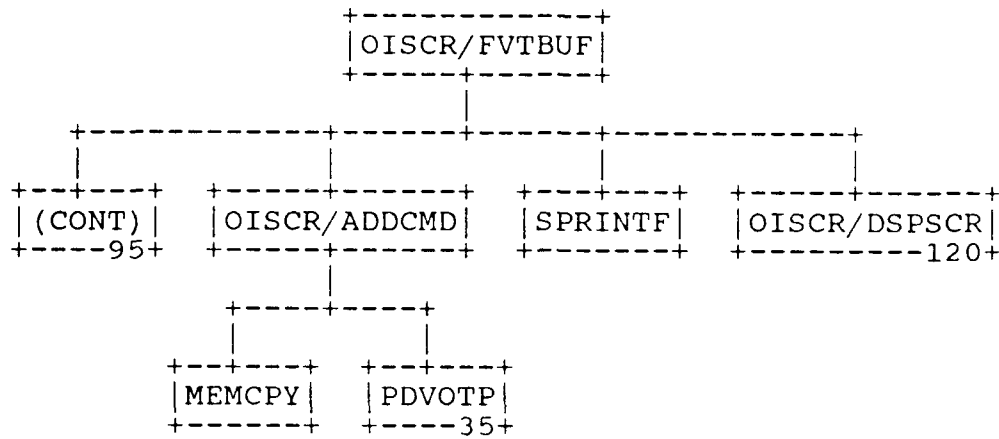
106



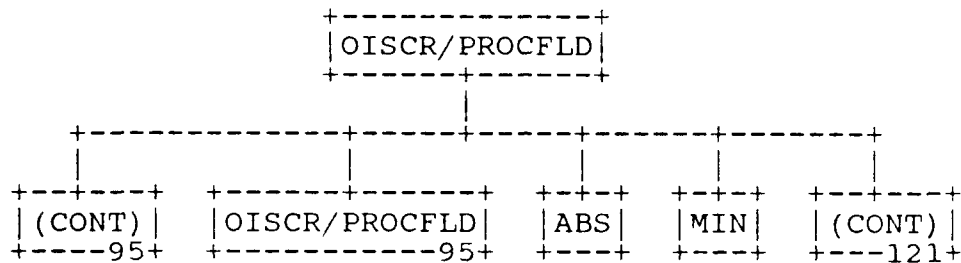
107



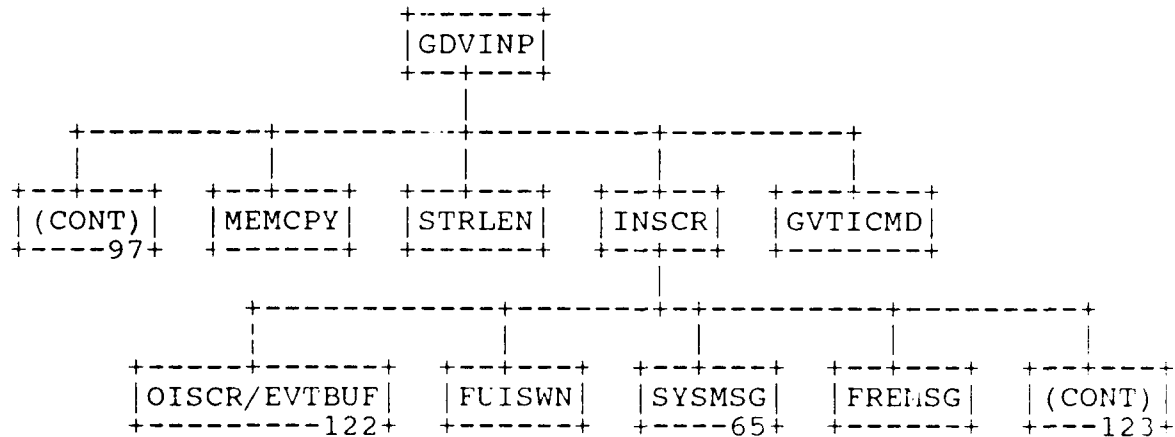
108



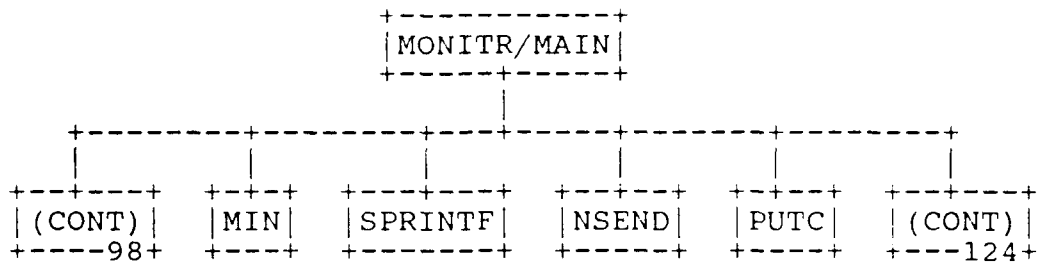
109



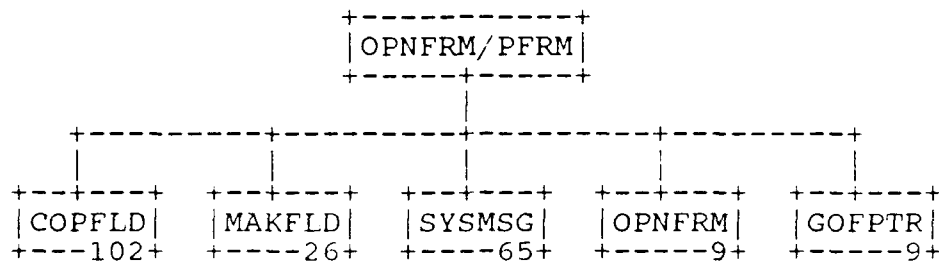
110



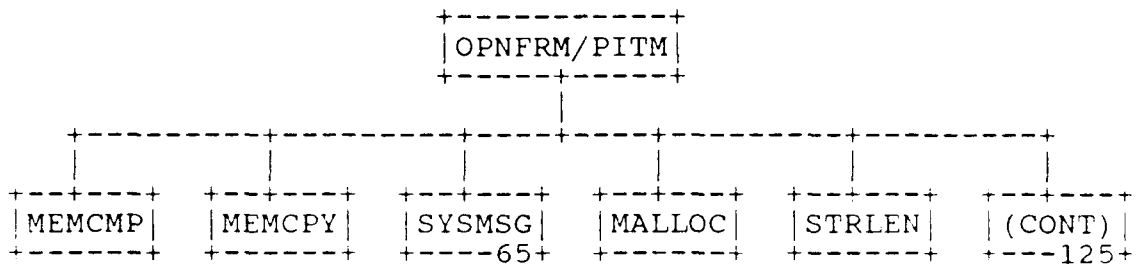
111



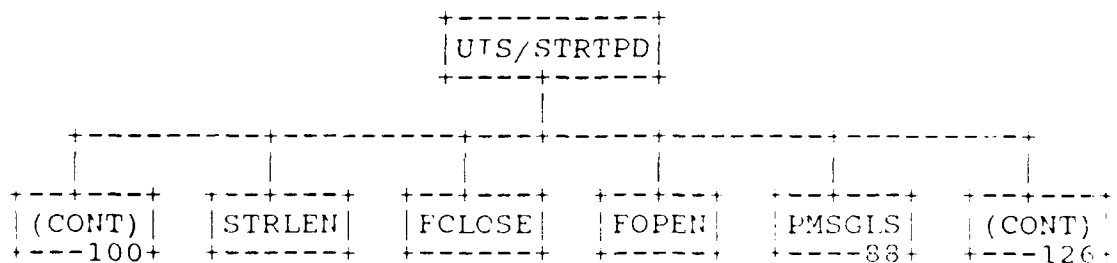
112



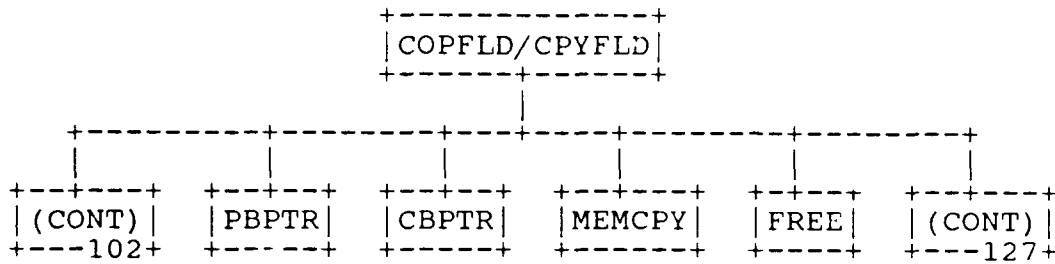
113



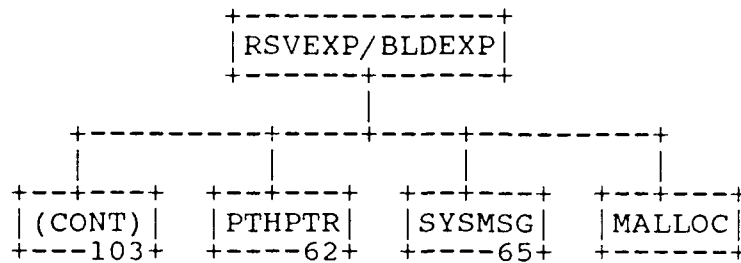
114



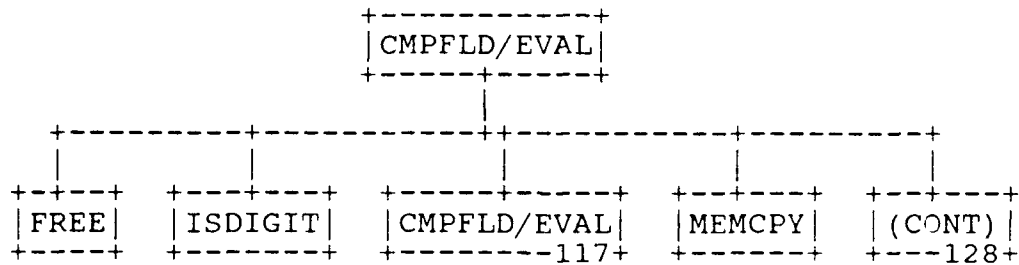
115



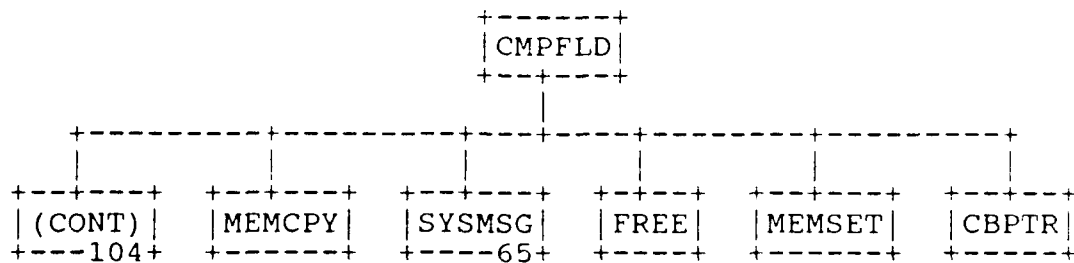
116



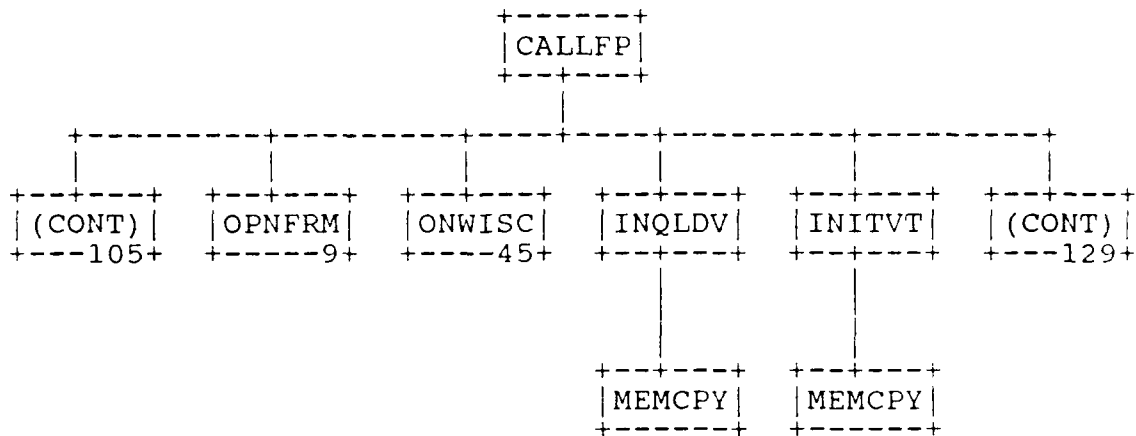
117



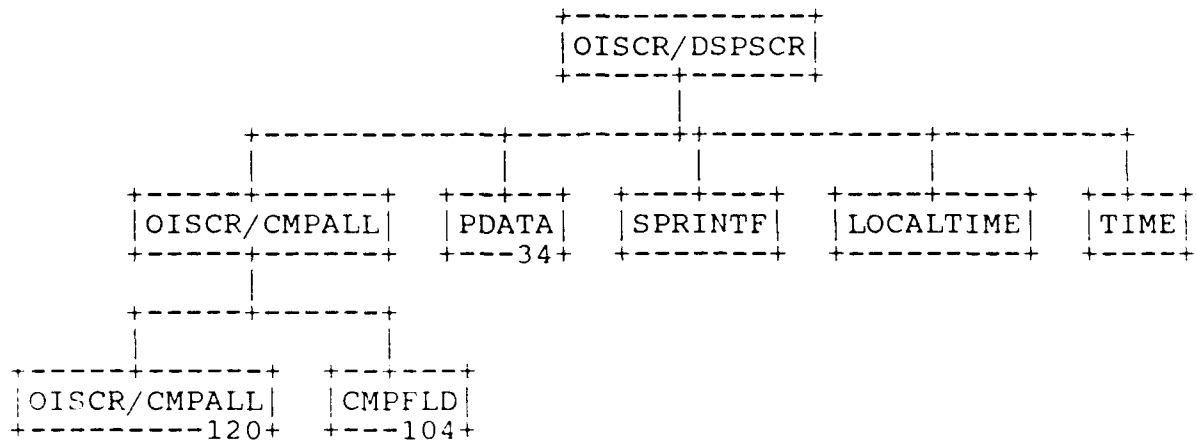
118



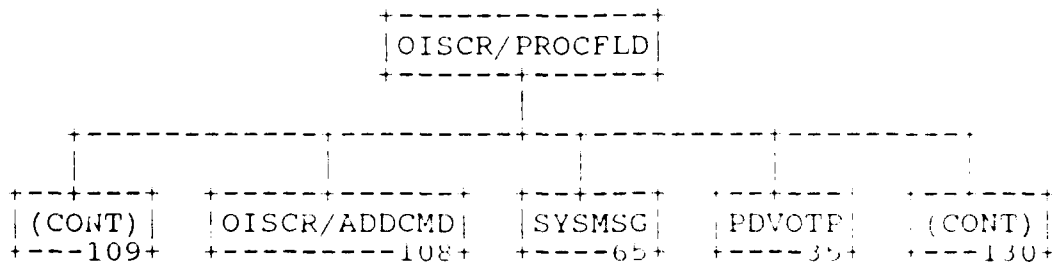
119



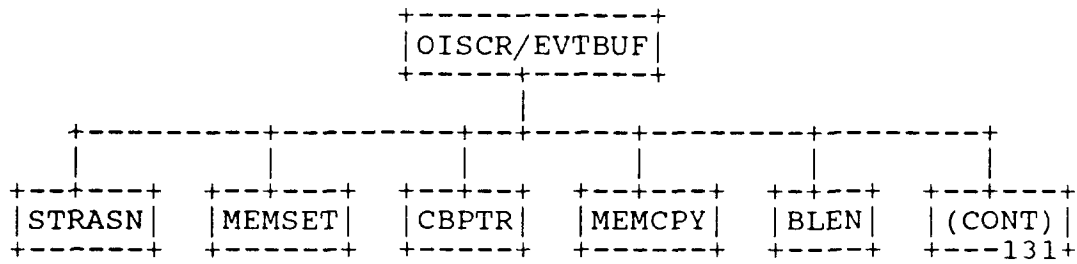
120



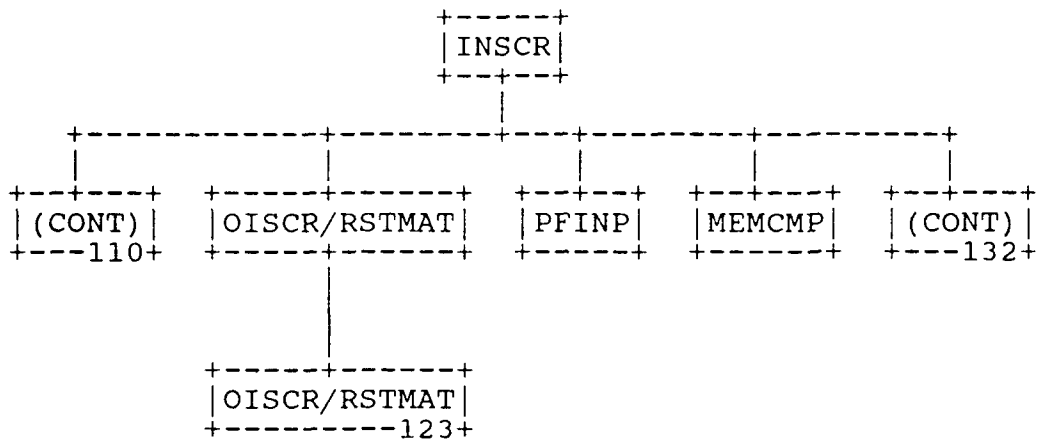
121



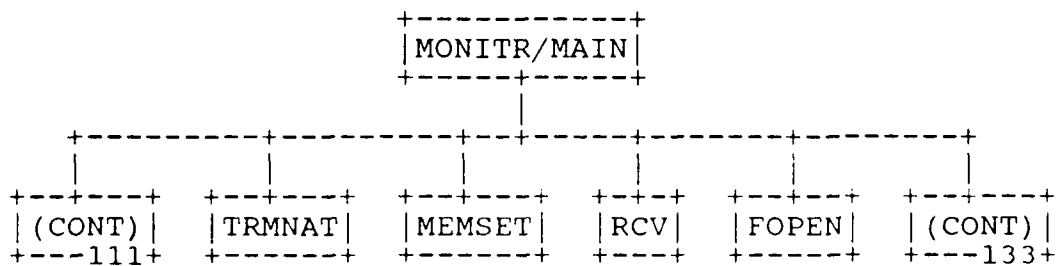
122



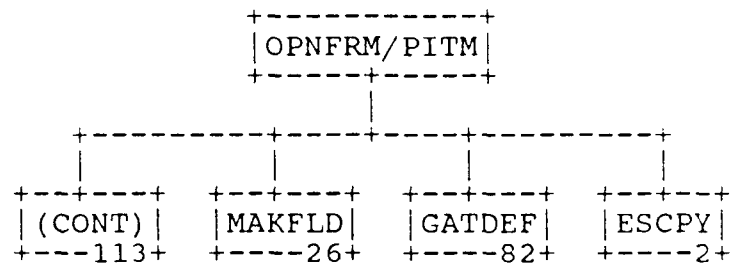
123



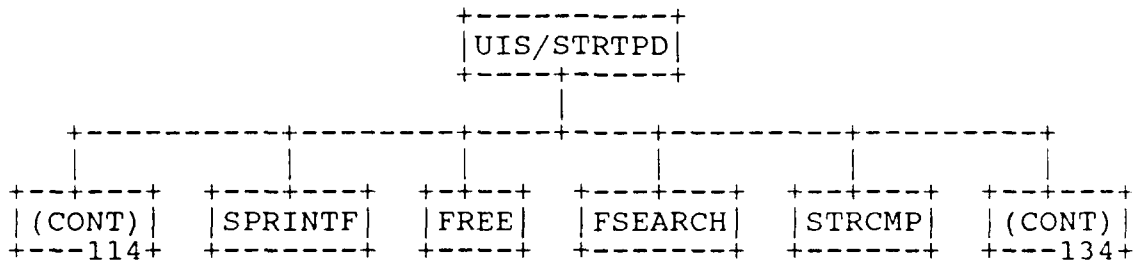
124



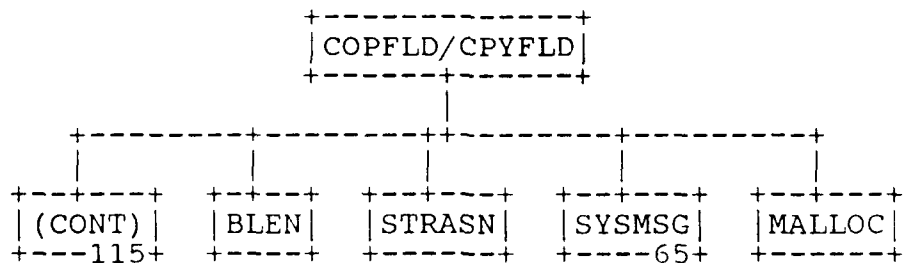
125



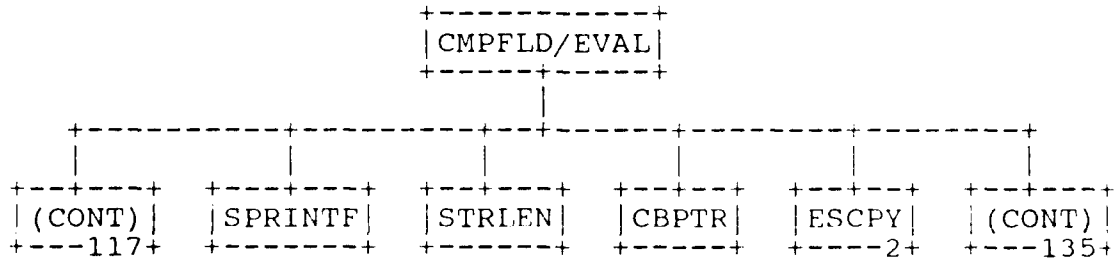
126



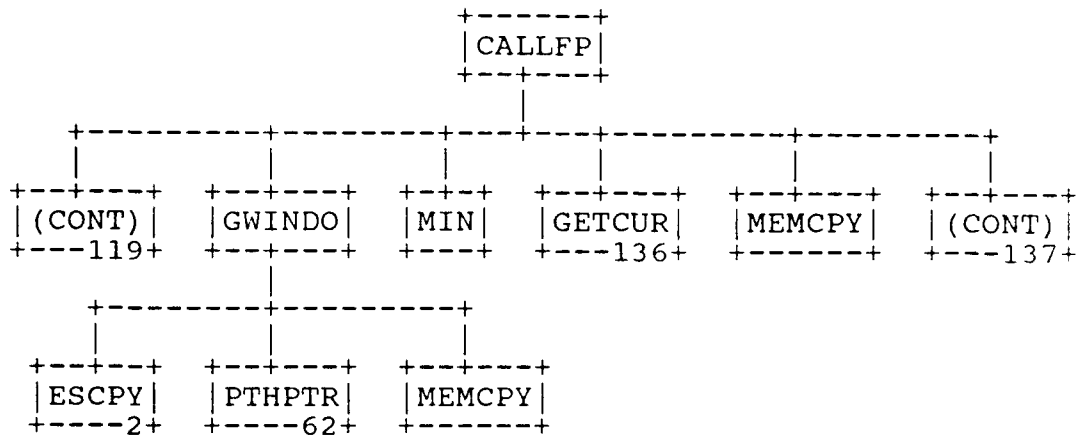
127



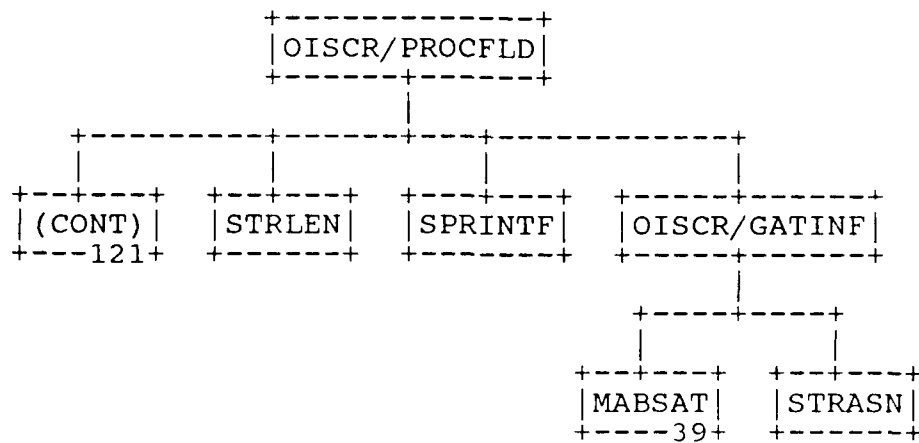
128



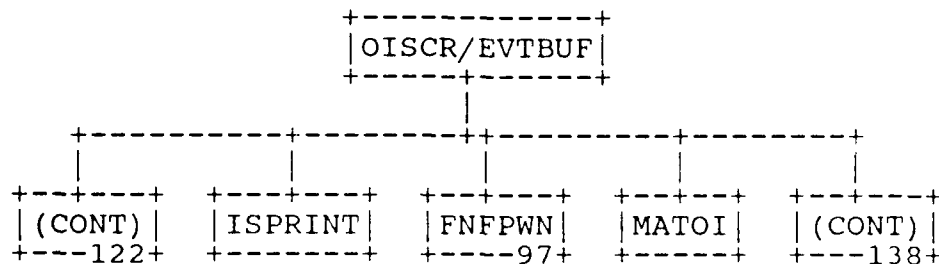
129



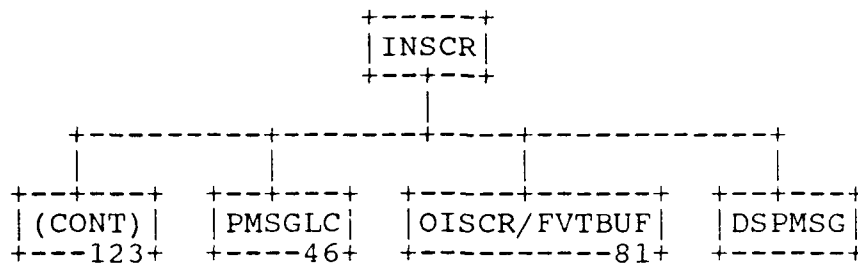
130



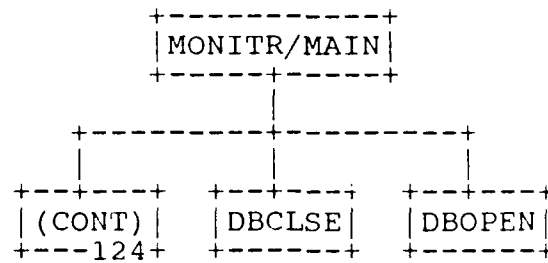
131



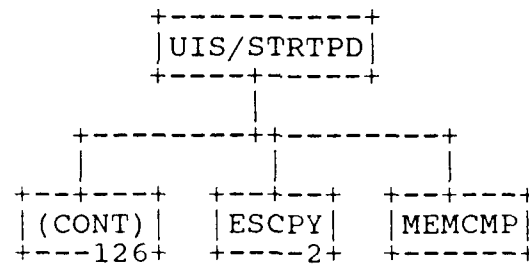
132



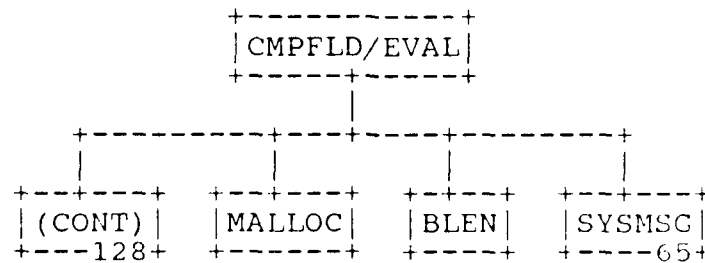
133



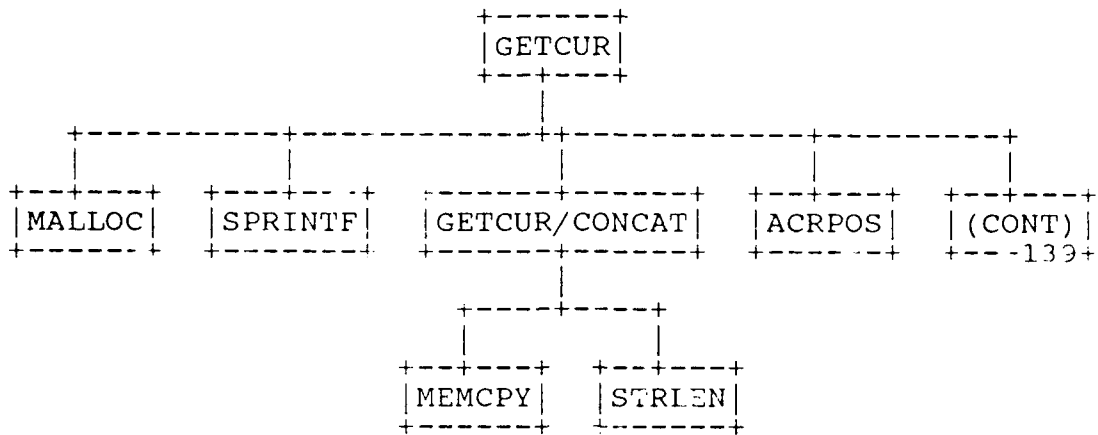
134



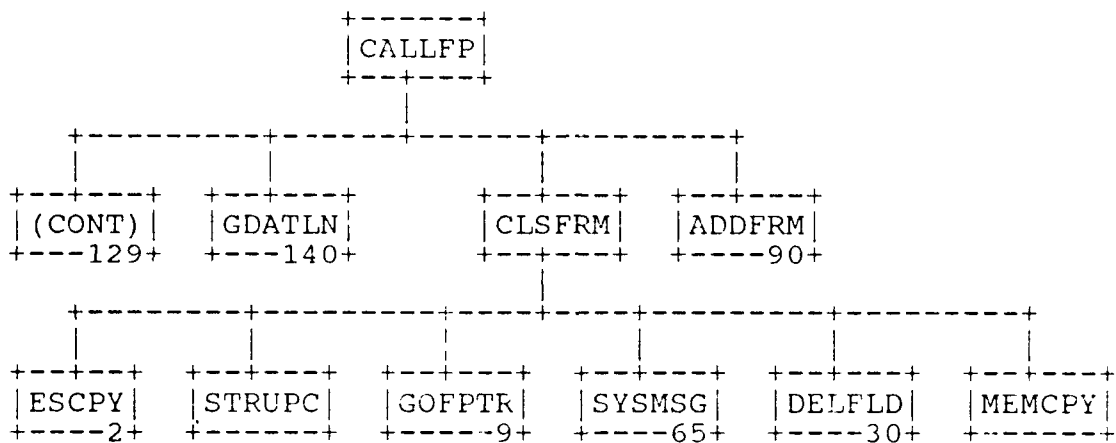
135



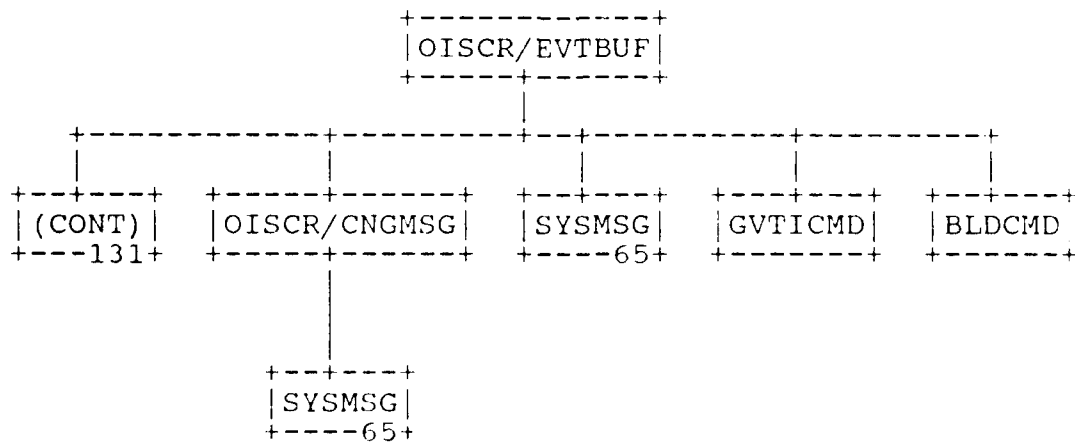
136



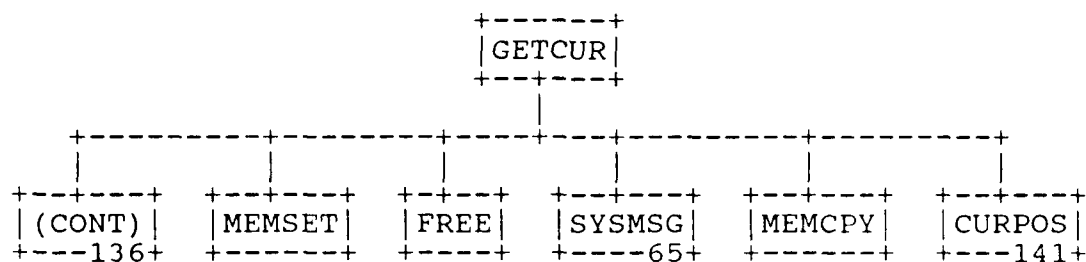
137



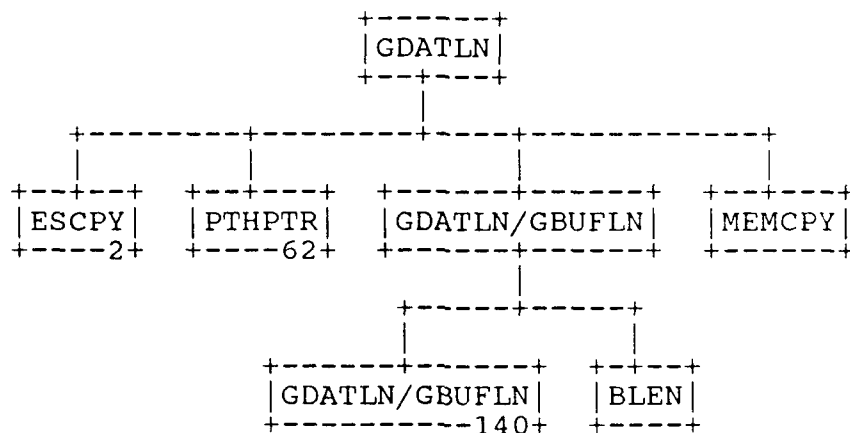
138



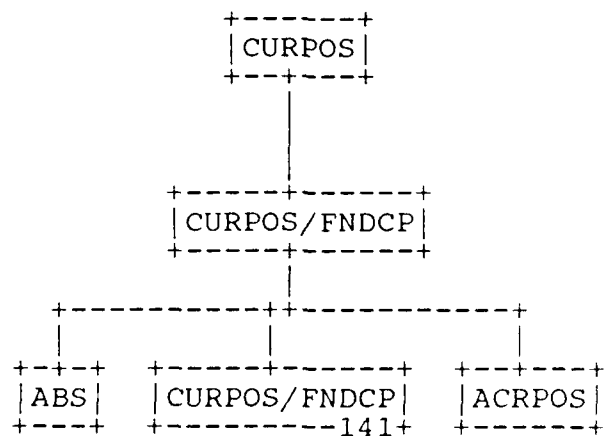
139



140



141



ABORT	FNDMSG/CODSCH...67	MATOI
ABS	FNDMSG/OUNSGF...46	MAX
ACRPOS	FNFPWN...97	MEMCMP
ADDELM.....78	FOPEN	MEMCPY
ADDFRM.....90	FPRINTF	MEMSET
ADJSTR.....6	FREAD	MIN
ATOI	FREE	MONITR/GETPD....66
BLDCMD	FREMSG	MONITR/MAIN.....2
BLFN	FSEARCH	NSEND
CALLFP.....14	FSEEK	OBIND
CALLOC	FTELL	ODFINN
CANITM.....104	FUISWN	OEXEC
CBIT	FWRITE	OFETCH
CBPTR	GATDEF.....82	OISCR/ADDCMD...108
CFREE	GDATA.....89	OISCR/CMPALL...120
CHGLDV.....60	GDATA/GETBUF...89	OISCR/CNGMSG...138
CHGPRC	GDATLN.....140	OISCR/DSPSCR...120
CLSFRM.....137	GDATLN/GBUFLN..140	OISCR/EVTBUF...122
CLSLDV.....77	GDVINP.....83	OISCR/FVTBUF...81
CMPFLD.....104	GETATT.....42	OISCR/GATINF...130
CMPFLD/EVAL...117	GETBAK.....59	OISCR/PROCFLD...95
COPFLD.....102	GETC	OISCR/PROCWIN...81
COPFLD/CPYFLD..102	GETCUR.....136	OISCR/RSTINP...63
COPFRM.....90	GETCUR/CONCAT..136	OISCR/RSTMAT...123
CURPOS.....141	GETW	OISCR/SETWIN...45
CURPOS/FNDLCP...141	GOFPTR.....9	ONWISC.....45
DBCFCNC.....87	GPAGE.....41	OPNFRM.....9
DBCLSE	GVTICMD	OPNFRM/BDBUFF...48
DBCOM	GVTINW	OPNFRM/BFLddb...70
DBCROL.....12	GWINDO.....129	OPNFRM/BRPNOD...47
DBCUPR	INITAL	OPNFRM/BTBUF...49
DBGAPD	INITVT.....119	OPNFRM/PARY.....85
DBOPEN	INQLDV.....119	OPNFRM/PDREC...72
DELFLD.....30	INSCR.....110	OPNFRM/PFREC...29
DELFLD/DELEXP...30	ISALNUM	OPNFRM/PFRM...112
DOATTR	ISCNTRL	OPNFRM/PITM...113
DOITEM	ISDIGIT	OPNFRM/PTREC...50
DOWIND	ISEND	OPNFRM/PWIN...99
DSPMSG	ISPRINT	OPNLDV.....105
ERRPRO	LOCALTIME	OSQL3
ESCPY.....2	MABSAT.....39	OUTSCR.....46
FCLOSE	MAKAP.....33	PARFQN.....40
FEOF	MAKFLD.....26	PBPTR
FERROR	MAKFPD.....43	PDATA.....34
FFBCA	MAKPD.....61	PDATA/PUTBUF...34
FNDFLD.....47	MAKUSR.....26	PDVOTP.....35
FNDMSG.....46	MALLOC	PFINP

PMSGLC.....46	STRLEN
PMSGLS.....88	STRNCMP
POSCUR.....19	STRNCPY
POSCUR/FNFITM...19	STRNLOC
PRINTF	STRNUPC
PRNAP.....4	STRRCHR
PRNDSP.....3	STRUPC
PRNFLD.....3	STUPFP.....26
PRNOPN.....1	SYMSG.....65
PRNPD.....4	TERMVT.....93
PRNUID.....4	TIME
PRNUSR.....5	TOUPPER
PTHPTR.....62	TRMDRV.....25
PTHPTR/ARRAY...107	TRMNAT
PTHPTR/FIELD...94	TRMUSR.....15
PTHPTR/FORM...94	UIS.....7
PTHPTR/FOUND...94	UIS/FLWINF.....18
PTHPTR/ITEM...94	UIS/FLWNST.....11
PTHPTR/WINDOW..106	UIS/PRCINP.....17
PUTATT.....39	UIS/PRCWND.....10
PUTATT/AABSAT...39	UIS/STRTAP.....33
PUTBAK.....38	UIS/STRTPD.....100
PUTC	ULKFPD
PUTCUR.....19	
PUTLOC.....14	
PUTW	
RCV	
REWIND	
RMVAP.....13	
RMVFPD.....21	
RMVPAG.....76	
RMVPD.....25	
RPLFRM.....23	
RSVATT.....37	
RSVATT/RSVRST...37	
RSVEXP.....92	
RSVEXP/BLDEXP..103	
SBIT	
SFPDAP.....8	
SIGABT	
SNDVTI	
SPRINTF	
STRASN	
STRCAT	
STRCHR	
STRCMP	
STRCPY	

3.11 Program Listings Comments

This information is contained in the Module Descriptions in section 3.10.

SECTION 4

QUALITY ASSURANCE PROVISIONS

4.1 Introduction and Definitions

"Testing" is a systematic process that may be preplanned and explicitly stated. Test techniques and procedures may be defined in advance, and a sequence of test steps may be specified. "Debugging" is the process of isolation and correction of the cause of an error.

"Antibugging" is defined as the philosophy of writing programs in such a way as to make bugs less likely to occur and when they do occur, to make them more noticeable to the programmer and the user. In other words, as much error checking as is practical and possible in each routine should be performed.

4.2 Computer Programming Test and Evaluation

The quality assurance provisions for test consists of the normal testing techniques that are accomplished during the construction process. They consist of design and code walk-throughs, unit testing, and integration testing. These tests are performed by the design team. Structured design, design walk-through and the incorporation of "antibugging" facilitate this testing by exposing and addressing problem areas before they become coded "bugs."